

4.12

VALI Help

The information contained herein is subject to change without notice.
Revisions may be issued to advise of such changes and/or additions.

VALI Help

Table of Contents

CHAPTER 1: WHAT'S NEW	23
SECTION 1: USER'S GUIDE ORGANISATION	23
SECTION 2: NEW FEATURES IN RELEASE 4.12	24
INSTALLATION	24
UPGRADING MODELS FROM RELEASE 4.11 TO RELEASE 4.12	24
KNOWN ISSUES	24
COMPATIBILITY	25
NEW FEATURES IN VALIMODELLER AND VALIENGINE	26
NEW FEATURES IN VALISCHEDULER AND VALISERVICE	26
CHAPTER 2: INTRODUCTION TO VALI	29
SECTION 1: DVR PRINCIPLES	29
INTRODUCTION	29
WHAT IS VALI ?	29
WHAT ARE THE BENEFITS OF DVR ?	30
THE BASICS OF DATA VALIDATION AND RECONCILIATION	30
SECTION 2: VALIMODELLER ENVIRONMENT	32
VALIMODELLER COMPONENTS	32
THE MODEL FILE (BLS) AND OTHER FILES	32
SECTION 3: VALI OBJECTS	34
PFD	34
UNIT	35
STREAM	35
THERMOD	36
REACTION	36
COMPOUND	37
TAG	37
AN EXAMPLE: THE LIQUID-VAPOUR SEPARATOR	39
SECTION 4: PROCESS UNIT EQUATIONS	41
CONTROLLING THE SET OF GENERATED EQUATIONS	42
BLACK BOXES AND MIXERS: BBXVAL, NETVAL, EFFVAL, CUTVAL	45
HEAT-EXCHANGERS: EEXVAL, HEXVAL, PRHVAL, SATVAL, WALVAL, ECOVAL	47

VALI Help

COMPRESSORS AND TURBINES: COMVAL AND TURVAL.....	50
PUMPS: PUMVAL.....	51
CHEMICAL REACTORS: BATVAL, BWHVAL, RCTVAL, RLVVAL, RKPVAL, RKIVAL, CND-HNO ₃ , HEX-HNO ₃ , RCT-HNO ₃	52
SEPARATORS: LVEVAL, LLEVAL, LLVVAL, WLEVAL, WLVVAL.....	53
DISTILLATION, ABSORPTION AND STRIPPING COLUMNS: COLVAL, DISVAL, CRUVAL.....	53
STORAGE TANKS: TNKVAL.....	54
USER DEFINED EQUATIONS AND CONTROLLERS: OPXVAL, FLXVAL, VARVAL, EQUVAL.....	54
VALVES AND PRESSURE DROP: DPVAL, PIPEVAL.....	55
ON/OFF PROPAGATION.....	55
SECTION 5: PROCESS MEASUREMENTS.....	57
MEASUREMENT INACCURACY (OR UNCERTAINTY).....	57
COVARIANCE BETWEEN MEASUREMENT ERRORS.....	59
MEASUREMENT REDUNDANCY.....	59
HANDLING PRESSURE MEASUREMENTS.....	60
AVAILABLE STREAM MEASUREMENTS.....	60
AVAILABLE UNIT MEASUREMENTS.....	68
EFFECT OF BOUNDS.....	69
VALIDATED ACCURACY AND COVARIANCE.....	70
GROSS ERROR DETECTION AND ELIMINATION.....	71
SECTION 6: PETROLEUM REFINING APPLICATIONS.....	73
THE CONCEPT OF PSEUDO-COMPOUND.....	73
DATA VALIDATION OF DISTILLATION CURVES.....	73
USING THE CRUVAL UNIT.....	74
USING DISTILLATION CURVES MEASUREMENTS.....	74
SECTION 7: COMPATIBILITY WITH VDI-2048.....	76
THE VDI-2048 GUIDELINE.....	76
THE VDI MODE OF VALI.....	76
COMPLIANCE TO THE VDI-2048 GUIDELINE.....	76
CHAPTER 3: THERMODYNAMIC CALCULATION METHODS.....	79
SECTION 1: INTRODUCTION.....	79
SECTION 2: BELSIM CALCULATION METHODS.....	80

VALI Help

FUGACITY CALCULATION METHODS FOR THE VAPOUR PHASE.....	81
FUGACITY CALCULATION METHODS FOR THE LIQUID PHASE	81
CALCULATION METHODS FOR ENTHALPY AND ENTROPY	83
CALCULATION METHODS FOR MOLAR VOLUME.....	84
CALCULATION METHODS FOR THERMAL CONDUCTIVITY	85
CALCULATION METHODS FOR VISCOSITY.....	86
CALCULATION METHODS FOR SOLID COMPOUNDS.....	86
PHASE EQUILIBRIUM CALCULATIONS	86
SECTION 3: FUNDAMENTAL PROPERTIES OF CHEMICAL COMPOUNDS	88
AVAILABLE DATA BASES.....	88
DATA NEEDED FOR LIQUIDS AND VAPOURS.....	90
SECTION 4: INTERACTION PARAMETERS	97
EQUATIONS OF STATE.....	98
VIRIAL EQUATION OF STATE	98
NRTL AND UNIQUAC METHODS.....	99
QUADRATIC HEAT OF MIXING.....	100
SECTION 5: CALCULATION METHODS FOR TYPICAL MIXTURES.....	101
WATER.....	101
NONPOLAR HYDROCARBON MIXTURES WITH OR WITHOUT WATER.....	101
OXYGENATED COMPOUND MIXTURES.....	103
SECTION 6: REFERENCES.....	105
SECTION 7: FLUIDS DATA BASE.....	108
SECTION 8: SOLIDS DATA BASE	138
SECTION 9: USER-DEFINED COMPOUNDS.....	160
STARTING NEWCOMP.EXE.....	160
ADDING A NEW COMPOUND.....	161
DELETING A COMPOUND.....	162
EDITING COMPOUND DEFINITION	162
EDIT COMPOUND DATA.....	163
SECTION 10: USING CO-COMPLIANT THERMODYNAMIC PLUGS.....	165
THE PRINCIPLES.....	165
DEFINING A CAPE-OPEN COMPLIANT THERMOD IN THE MODEL FILE	165

VALI Help

REVIEWING THE PROPERTIES OF A CO COMPLIANT THERMOD.....	166
SECTION 11: UNCERTAINTY OF THERMODYNAMICS	168
INTRODUCTION.....	168
CALCULATION PROCEDURE.....	168
CHAPTER 4: HOW TO BUILD YOUR DATA VALIDATION MODEL	173
SECTION 1: INTRODUCTION.....	173
SECTION 2: HOW TO BUILD A VALIDATION MODEL	174
PRELIMINARY WORK.....	174
DEFINING THE MODEL WITH VALIMODELLER.....	177
SECTION 3: INTRODUCING VALIMODELLER.....	179
WINDOWS STANDARD.....	179
INFORMATION COHERENCY.....	180
STARTING VALIMODELLER.....	181
THE BROWSE TREE	182
MENU ITEMS.....	184
EDIT MODE.....	188
TOOLBAR AND KEYBOARD SHORT-CUTS	189
SHOWING UNIT AND STREAM NAMES AND RESULT BOXES.....	191
CHANGING CONFIGURATION OPTIONS.....	191
MODEL MEASUREMENTS PAGE.....	197
THE FIND FACILITY.....	198
RENAMING OBJECTS.....	199
SELECTING OBJECTS.....	200
SECTION 4: THE MODEL FILES.....	201
CONTENTS OF A MODEL FILE.....	201
CREATING A NEW MODEL FILE.....	201
IMPORTING MODELS FROM AN OTHER MODEL FILE	202
MODEL VERSIONING AND DOCUMENTATION.....	204
THE DEFAULT PLANT.....	205
SECTION 5: THE PROCESS FLOW DIAGRAMS (PFD).....	207
CREATING A NEW VALI PFD.....	207
OPENING A PFD.....	208

VALI Help

RENAMING A PFD.....	208
DELETING A PFD.....	208
MOVING A PFD TO ANOTHER PFD.....	209
CHANGING THE STATUS OF A PFD (ON, OFF, OUT).....	209
USING EXTERNAL PFDs.....	210
ZOOMING A PFD.....	211
PRINTING A PFD.....	211
SECTION 6: THE UNITS.....	212
UNIT STATUS : ON/OFF/OUT	212
UNIT NOTEBOOK.....	212
CREATING A UNIT	220
MOVING AND SIZING UNITS.....	222
DELETING UNITS.....	223
RENAMING UNITS.....	223
CREATING A FLEX PROGRAM FROM AN EXTERNAL FILE.....	223
SECTION 7: THE STREAMS	225
STREAM NOTEBOOK.....	225
CREATING A STREAM.....	230
REDRAWING AND RECONNECTING STREAMS.....	232
DELETING STREAMS.....	233
RENAMING STREAMS	233
NAVIGATING BETWEEN PFDs.....	234
SECTION 8: THE TAGS.....	235
SETS OF VALUES ASSOCIATED TO A TAG.....	236
CREATING NEW TAGS.....	238
EDITING TAGS.....	239
COPYING TAGS FROM EXISTING OBJECTS.....	243
MOVING TAGS FROM EXISTING OBJECTS.....	244
VIEWING AND EDITING MEASUREMENTS VALUES.....	245
USING CONTROL STRINGS	253
SECTION 9: THE THERMODS	257
ACCESSING THE THERMODS NOTEBOOKS	257

VALI Help

MODIFYING THERMODS.....	258
CREATING THERMODS.....	259
RENAMING THERMODS.....	260
DELETING THERMODS.....	260
SECTION 10: THE COMPOUNDS.....	261
ADDING COMPOUNDS TO A MODEL FILE.....	262
REMOVING COMPOUNDS FROM A MODEL FILE.....	264
RENAMING COMPOUNDS IN A MODEL FILE.....	264
SECTION 11: THE REACTIONS.....	265
ACCESSING REACTIONS NOTEBOOK.....	265
MODIFYING THE STOICHIOMETRY OF REACTIONS.....	266
MODIFYING THE KINETICS OF REACTIONS.....	267
CREATING REACTIONS.....	269
RENAMING REACTIONS.....	269
DELETING REACTIONS.....	269
SECTION 12: HANDLING PHYSICAL UNITS.....	270
CHANGING PHYSICAL UNITS.....	270
CREATING NEW PHYSICAL UNITS.....	271
SECTION 13: RESULT BOXES.....	272
RESULT BOXES NOTEBOOK.....	272
CREATING RESULT BOXES.....	274
DELETING RESULT BOXES.....	274
SECTION 14: A PRACTICAL EXAMPLE.....	275
THE EXAMPLE : A GAS TURBINE.....	275
PRELIMINARY WORK.....	277
DEFINING THE MODEL WITH VALIMODELLER.....	277
CHAPTER 5: HOW TO RUN YOUR DATA VALIDATION MODEL.....	289
SECTION 1: INTRODUCTION.....	289
SECTION 2: RUNNING THE MODEL.....	290
STARTING VALIENGINE FROM VALIMODELLER GUI.....	291
INTERACTING WITH VALIENGINE.....	296
BROWSING RESULT FILES.....	298

VALI Help

LOADING VALI RESULTS BACK IN VALIMODELLER GUI	302
USING VALI IN BATCH MODE.....	303
SECTION 3: USING TAG FILES.....	306
GENERAL SYNTAX OF A TAG FILE.....	306
TAG DEFINITION.....	307
CONTROL STRINGS.....	308
DEFINING DUMMY TAGS.....	308
EXAMPLE OF A TAG FILE.....	309
SECTION 4: USING MEASUREMENTS FILES.....	310
SYNTAX OF A MEASUREMENTS FILE.....	310
CHANGING VALUES OF TAGS.....	315
DESCRIPTIVE INFORMATION ABOUT THE RUN.....	317
SETTING THE REFERENCE ATMOSPHERIC PRESSURE.....	317
DEFINING BOUNDS ON TAGS	318
DEFINING BOUND CONTROL.....	318
DEFINING FILTER SETTINGS.....	319
DEFINING COMPENSATION SETTINGS.....	321
DEFINING COVARIANCES BETWEEN TAGS.....	322
UNIT PARAMETERS AND INITIAL VALUES OF VARIABLES	323
PRODUCT ATTRIBUTES.....	324
PRODUCT BALANCES AND IMBALANCES	325
COMMENTS OF OBJECTS.....	327
SETTING UNITS, STREAMS OR PRODUCT "OFF".....	328
SETTING UNITS OR PRODUCTS "OUT".....	329
THERMOD BINARY PARAMETERS.....	329
COMPOUND DATA	330
ELECTRO-NEUTRALITY CONSTRAINTS.....	330
OUTPUT AND ARCHIVE FILES.....	331
WRITING TAG VALUES IN THE REPORT	334
WRITING VARIOUS INFORMATION ITEMS IN THE REPORT	334
WRITING UNIT, STREAM, COMPOUND AND THERMOD VALUES	336
WRITING TABLES.....	340

VALI Help

USING IOFILE.EXE TO PROCESS MEASUREMENT FILES	350
USING IOFILE.EXE TO CHANGE DEFAULT VALUES	350
EXAMPLE OF MEASUREMENT FILES.....	352
SECTION 5: THE SENSITIVITY ANALYSIS.....	354
PRELIMINARY REMARKS.....	355
REPORTS OF THE SENSITIVITY ANALYSIS.....	355
HOW TO PERFORM A SENSITIVITY ANALYSIS.....	360
THEORETICAL BACKGROUND.....	364
SECTION 6: TRICKS AND TIPS.....	374
HOW TO INITIALIZE A MODEL ?.....	374
SOLVING CONVERGENCE PROBLEMS.....	375
HOW TO CHECK THE COHERENCE OF A MODEL ?.....	376
HOW TO DEAL WITH CLOSED LOOPS ?.....	376
DEALING WITH NEGATIVE FLOW RATES.....	376
HOW TO SWITCH BETWEEN OPERATING POINTS?	376
CUSTOMIZING REPORTS.....	377
USER-DEFINED PHYSICAL UNITS.....	377
SECTION 7: VALI ERROR MESSAGES	380
VALI ERROR HANDLING FACILITY.....	380
ERRORS PRODUCED IN THE "MODEL ANALYSIS" PHASE	382
ERRORS PRODUCED IN THE "MODEL RESOLUTION" PHASE.....	386
CHAPTER 6: DATA VALIDATION UNITS.....	391
SECTION 1: SELECTING UNIT TYPES.....	391
SECTION 2: ATMOPRES: SETTING ATMOSPHERIC PRESSURE.....	395
SECTION 3: BATVAL: REACTOR WITH ATOMIC BALANCES.....	396
SECTION 4: BBXVAL: BLACK BOX WITH MOLAR BALANCES.....	402
SECTION 5: BNDVAL: DEFAULT BOUNDS ON VARIABLES.....	409
SECTION 6: BWHVAL: EQUILIBRATED REACTOR WITH ATOMIC BALANCES	412
SECTION 7: CND-HNO3: NITROUS GAS CONDENSER.....	419
SECTION 8: COLVAL: DISTILLATION COLUMN.....	424
SECTION 9: COMVAL: GAS COMPRESSOR.....	432
SECTION 10: CRUOPT: OPTIONS FOR CRUDE FRACTIONS IDENTIFICATION.....	436

VALI Help

SECTION 11: CRUVAL: CRUDE FRACTIONS IDENTIFICATION.....	440
SECTION 12: CUTOFF: OPTIONS FOR GENERIC CUTOFF.....	451
SECTION 13: CUTVAL: TEAR UNIT FOR LOOPS.....	453
SECTION 14: DISVAL: TRAY-BY-TRAY DISTILLATION COLUMN.....	458
SECTION 15: DPVAL: PRESSURE DROP (VALVE).....	463
SECTION 16: ECOVAL: BOILER ELEMENT.....	474
SECTION 17: EEXVAL: HEAT EXCHANGER.....	475
SECTION 18: EFFVAL: EFFICACITY AND LOSSES.....	481
SECTION 19: EQUVAL: COMPOSITION EQUALITY.....	483
SECTION 20: FLXVAL: USER DEFINED EQUATIONS.....	488
SECTION 21: FLEXLIB: USER DEFINED FUNCTIONS.....	494
SECTION 22: GROSSERR: GROSS ERRORS ELIMINATION.....	495
SECTION 23: HEX-HNO3: NITROUS GAS COOLER.....	500
SECTION 24: HEXVAL: RIGOROUS HEAT EXCHANGER.....	505
SECTION 25: LLEVAL: LIQUID-LIQUID-SEPARATOR.....	519
SECTION 26: LLVVAL : LIQUID-LIQUID-VAPOR SEPARATOR.....	523
SECTION 27: LOFUNIT (PFD): LIST OF UNITS.....	527
SECTION 28: LVEVAL : LIQUID-VAPOR SEPARATOR.....	528
SECTION 29: NETVAL : PIPES NETWORK FOR MASS BALANCE.....	533
SECTION 30: OPTVAL : VARIOUS VALI PARAMETERS.....	539
SECTION 31: OPXVAL : USER DEFINED EQUATIONS	546
SECTION 32: PIPEVAL : PIPELINES AND WELLS.....	551
SECTION 33: PRECIS: PRECISION OF CALCULATIONS.....	565
SECTION 34: PRHVAL : PREHEATER WITH CONDENSATION.....	571
SECTION 35: PUMVAL : LIQUID PUMP.....	580
SECTION 36: RCT-HNO3 : NH3 OXIDATION	588
SECTION 37: RCTVAL : REACTOR WITH DEFINED REACTIONS	592
SECTION 38: REPVAL : REPORT SETTINGS.....	598
SECTION 39: RKIVAL : REACTOR WITH KINETICS	609
SECTION 40: RKPVAL : EQUILIBRATED REACTIONS	614
SECTION 41: RLVVAL : REACTOR IN L/V EQUILIBRIUM.....	621
SECTION 42: SATVAL : SATURATION UNIT	624

VALI Help

SECTION 43: SPLVAL : SPLITTER.....	629
SECTION 44: TNKVAL : STORAGE TANK.....	632
SECTION 45: TURVAL : TURBINE.....	641
SECTION 46: VARVAL : USER DEFINED VARIABLES.....	646
SECTION 47: WALVAL : HEAT LOSSES THROUGH UNIT WALLS.....	647
SECTION 48: WLEVAL : WATER-LIQUID ORGANIC SEPARATOR.....	652
SECTION 49: WLVVAL : LIQUID-LIQUID-VAPOR SEPARATOR.....	655
CHAPTER 7: FLEX LANGUAGE DESCRIPTION.....	661
SECTION 1: INTRODUCTION.....	661
WHAT YOU CAN DO WITH FLEX.....	661
INTERPRETED OR COMPILED.....	662
SECTION 2: FLEX LANGUAGE OBJECTS.....	663
SECTION 3: LANGUAGE ELEMENTS.....	664
SEPARATORS.....	664
CONSTANTS.....	664
VARIABLES.....	665
OPERATORS AND INTERNAL FUNCTIONS.....	667
OPERATIONS EXECUTION ORDER.....	669
RESERVED KEYWORDS.....	669
SECTION 4: EXECUTABLE INSTRUCTIONS.....	671
AFFECTATION.....	671
READ.....	671
WRITE.....	671
WHILE BLOCK.....	672
UNTIL BLOCK.....	672
IF BLOCK.....	673
END.....	673
RETURN.....	673
SYNTAX OF EXECUTABLE INSTRUCTIONS.....	674
SECTION 5: FLEX FUNCTIONS.....	675
USING FLEX FUNCTIONS.....	675
FLEX INTERNAL FUNCTIONS.....	676

VALI Help

USER-DEFINED FUNCTIONS.....	678
SECTION 6: EXAMPLE OF A FLEX PROGRAMME.....	680
SECTION 7: VALI IMPLEMENTATION OF FLEX.....	681
INTRODUCTION.....	681
USER DEFINED MEASUREMENTS.....	682
FLEX FUNCTIONS TO SELECT VARIABLES AND DATA.....	684
FLEX FUNCTIONS FOR THERMODYNAMIC CALCULATIONS.....	694
FLEX FUNCTIONS USED IN FILTERING.....	698
SECTION 8: USING FLEX FOR DEFINING KINETICS.....	701
CHAPTER 8: SIMULATION UNITS.....	705
SECTION 1: COL010 : DISTILLATION COLUMN.....	705
SECTION 2: CTL010 : SINGLE VARIABLE CONTROLLER.....	708
SECTION 3: CTL020 : MULTI-VARIABLES CONTROLLER.....	712
SECTION 4: CUT010 : UTILITY TO CUT STREAMS.....	716
SECTION 5: FUR010 : FURNACE.....	718
SECTION 6: HTX010 : HEAT TRANSFER.....	721
SECTION 7: HTX040 : HEAT EXCHANGER.....	725
SECTION 8: HTX050 : FURNACE TO HEAT UP A GAS.....	729
SECTION 9: HTX060 : BOILER WITH ASSOCIATED DRUM.....	734
SECTION 10: HTX070 : SHELL AND TUBES HEAT EXCHANGER.....	738
SECTION 11: HTX100 : HEAT EXCHANGER (ONE SIDE).....	743
SECTION 12: HTX110 : STREAM STATE TOOLBOX.....	746
SECTION 13: INCREM : LOOP CONTROLLER.....	748
SECTION 14: MAT010 : CONVERGENCE PROMOTER.....	751
SECTION 15: MAT020 : CONVERGENCE PROMOTER.....	755
SECTION 16: MEC010 : GAS COMPRESSOR.....	758
SECTION 17: MEC030 : TURBINE.....	760
SECTION 18: MEC050 : LIQUID PUMP.....	762
SECTION 19: MEC110 : GAS COMPRESSOR.....	764
SECTION 20: MEC130 : GAS COMPRESSOR.....	768
SECTION 21: MIS010 : UTILITY MODEL.....	773
SECTION 22: MIX010 : MATERIAL STREAMS MIXER.....	774

VALI Help

SECTION 23: MIX020 : MECHANICAL/THERMAL STREAMS MIXER.....	776
SECTION 24: PERMEA : PERMEATOR.....	777
SECTION 25: OPT070 : OPTIMIZER.....	781
SECTION 26: RCT010 : CHEMICAL REACTOR.....	785
SECTION 27: RCT110 : NH3 SYNTHESIS REACTOR.....	789
SECTION 28: RCT141 : FURNACE TYPE CHEMICAL REACTOR.....	792
SECTION 29: RCT201 : CHEMICAL REACTOR.....	798
SECTION 30: RCT210 : CHEMICAL REACTOR.....	802
SECTION 31: RCT300 : INTEGRAL PLUG FLOW REACTOR.....	806
SECTION 32: RCT320 : COOLED PLUG FLOW REACTOR.....	811
SECTION 33: RCT400 : FLUID BED REACTOR.....	817
SECTION 34: RCT420 : FLUID BED REACTOR.....	818
SECTION 35: RCT500 : CSTR REACTOR.....	824
SECTION 36: REPORT : REPORT FILE CONTROLLER.....	829
SECTION 37: SEP010 : SPLITTER.....	831
SECTION 38: SEP020 : LIQUID - VAPOUR SEPARATOR.....	833
SECTION 39: SEP030 : BLACK BOX SEPARATOR.....	836
SECTION 40: SEP050 : SPLITTER.....	840
SECTION 41: SEP060 : LIQUID/LIQUID SEPARATOR.....	842
SECTION 42: SEP070 : THREE PHASE SEPARATOR (L/L/V).....	844
SECTION 43: SEP080 : SOLID/FLUID PHASE SEPARATOR.....	847
SECTION 44: SIMFLX : USER-DEFINED CALCULATIONS.....	849
DEFINING THE FLEX PROGRAM.....	849
DEFINING WHEN THE CODE MUST BE EXECUTED.....	851
FLEX DECLARATIONS OF VARIABLES.....	851
FLEX ROUTINE TO SELECT THERMODYNAMIC MODEL.....	852
FLEX SUBROUTINES TO SELECT VARIABLES.....	853
FLEX SUBROUTINES FOR THERMODYNAMIC CALCULATIONS.....	856
CHAPTER 9: VALIREPORT.....	863
SECTION 1: INTRODUCTION.....	863
WHAT CAN YOU DO WITH THIS APPLICATION?.....	863
REQUIREMENTS.....	864

VALI Help

SPECIAL FEATURES AND CONVENTIONS	864
SECTION 2: HANDLING VALIREPORT WORKBOOKS.....	865
CREATING A VALIREPORT WORKBOOK.....	865
OVERVIEW OF VALIREPORT.....	867
MOVING AND COPYING A VALIREPORT WORKBOOK.....	869
WORKING IN READ-ONLY MODE.....	869
SECTION 3: HANDLING DATABASES.....	870
ACCESS DATABASE.....	870
SQL-SERVER DATABASE.....	870
DATABASE STRUCTURE.....	873
SECTION 4: WORKING RUN BY RUN.....	874
LOAD RUN DATA FILES INTO VALIREPORT.....	874
STORE RUN DATA INTO A DATABASE.....	877
AUTOMATICALLY STORE RUNS INTO A DATABASE.....	877
RETRIEVE RUN STORED IN THE DATABASE.....	879
REMOVING RUNS FROM THE DATABASE.....	880
GROUPING TAGS BY UNITS	880
SECTION 5: TABLES AND CHARTS.....	882
CREATING A TABLE	882
CREATING A CHART.....	883
UPDATING TABLES AND CHARTS.....	883
SECTION 6: FLOW-SHEETS AND REPORTS	884
CREATING A NEW FLOW-SHEET.....	884
DRAWING THE FLOW-SHEET	884
INSERTING TAG REFERENCES: FSTAG.....	886
INSERTING AGGREGATED TAG REFERENCES: BLSTAGC.....	887
MODIFYING TAG REFERENCES.....	889
DELETING TAG REFERENCES.....	889
CHANGING COLOURS AND THRESHOLD VALUE.....	889
SECTION 7: WORKING DIRECTORY.....	891
SECTION 8: VALIDBLOADER APPLICATION.....	892
INI FILE FOR AN ACCESS DATABASE.....	893

VALI Help

INI FILE FOR A SQL SERVER DATABASE.....	893
ADDITIONAL SECTIONS IN THE INI FILE.....	894
VALIDBLOADER STATUS.....	895
SECTION 9: RUNNING VALI.....	896
CHAPTER 10: VALIANALYTICS.....	903
SECTION 1: INTRODUCTION.....	903
SECTION 2: INTEGRATION IN VALISUITE.....	906
SECTION 3: GENERALITIES.....	907
TOOLBAR.....	907
STATUS BAR.....	908
GENERAL FUNCTIONALITIES.....	909
SECTION 4: GLOBAL VIEW.....	911
GLOBAL TAGS TABLE.....	914
GAIN DISTRIBUTION CHART.....	916
GENERAL INFORMATION.....	917
GAIN IN UNCERTAINTY BY MEASUREMENT TYPE.....	919
SECTION 5: INDIVIDUAL VIEW.....	922
GLOBAL TAGS TABLE.....	926
TAG CHART.....	928
PRECEDENTS/DEPENDENTS TABLES.....	931
PRECEDENTS CHARTS.....	935
CHAPTER 11: LICENSEMANAGER.....	945
SECTION 1: INTRODUCTION.....	945
SECTION 2: BELSIM LICENSE SERVER.....	946
LICENSING PHILOSOPHY.....	946
USERS RIGHTS.....	947
HANDLERS AND TIME-OUT.....	948
TAGS.....	948
PLANTS.....	949
ValiManager and ValiPerformance.....	949
SUB- LICENSES.....	949
SECTION 3: BELSIM LICENSEMANAGER.....	954

VALI Help

STARTING LICENSEMANAGER.....	954
THE GENERAL PAGE.....	956
THE VALIMANAGER PAGE.....	958
THE VALIPERFORMANCE PAGE.....	960
THE TAG PAGE.....	962
THE PLANT PAGE.....	964
THE USER PAGE.....	965
THE MAINTENANCE PAGE.....	969
THE HANDLER PAGE.....	970
CHAPTER 12: VALIPERFORMANCE.....	973
SECTION 1: VALI Software Suite.....	973
ValiPerformance.....	973
VALI modules.....	975
CHAPTER 13: VALILINK.....	979
SECTION 1: INTRODUCTION.....	979
USING VALILINK TO GET DATA.....	979
USING VALILINK TO WRITE DATA.....	982
TAGNAMES VERSUS DBNAMES.....	983
SECTION 2: RUNNING VALILINK.....	984
STARTING VALILINK.....	984
SPECIFYING THE VALIDATION TIME WINDOW.....	984
VALILINK STATUS.....	985
SECTION 3: CONFIGURING VALILINK - VALILINKBUILDER.....	986
STARTING VALILINKBUILDER.....	986
GENERAL SETTINGS.....	989
READING AND WRITING ACTIONS.....	995
FIELDS RECAPITULATIVE TABLE.....	1004
SECTION 4: INPUT AND OUTPUT FILES FORMAT.....	1016
COMMON FORMATTING CONVENTIONS.....	1016
FORMAT OF THE INPUT FILES.....	1016
FORMAT OF THE OUTPUT FILES.....	1016
SECTION 5: CONTENTS OF THE LOG FILE.....	1018

VALI Help

.SECTION 6: PLUGS TECHNICAL INFORMATION.....	1019
PI.....	1019
IP21.....	1020
PLANTCONNECT.....	1025
MS-SQL.....	1025
ORACLE.....	1026
PHD.....	1031
MATRIKON OPC HDA SERVER.....	1032
CHAPTER 14: VALIFILTER AND VALIEVENT.....	1037
.SECTION 1: INTRODUCTION.....	1037
.SECTION 2: VALIFILTER.....	1038
FILTERING TAGS.....	1038
DEFINING THE FILTERS USING VALIMODELLER.....	1038
DEFINING THE FILTERS USING MEA FILES.....	1040
INTERACTION WITH CUTOFF AND CONTROL STRINGS.....	1041
REPORTS.....	1042
ADVANCED FILTERING STRATEGIES.....	1043
.SECTION 3: VALIEVENT.....	1044
OPERATION.....	1044
STARTING VALIEVENT.....	1044
CONFIGURING VALIEVENT.....	1046
CHAPTER 15: VALI ONLINE ENGINE.....	1051
.SECTION 1: INTRODUCTION.....	1051
.SECTION 2: RUNNING VALIONLINEENGINE.....	1052
STARTING VALIONLINEENGINE.....	1052
VALI INFORMATION FILE (pc.vif).....	1052
RUN STATUS.....	1053
.SECTION 3: RUNNING VALISCHEDULER.....	1054
STARTING VALISCHEDULER.....	1054
GENERAL PAGE.....	1055
ACQUISITION PAGE.....	1069
RECONCILIATION PAGE.....	1070

VALI Help

STORAGE PAGE.....	1073
EVENT PAGE.....	1074
BACKUP PAGE.....	1075
LOG FILE PAGE.....	1076
EMAIL PAGE.....	1078
.SECTION 4: VALISERVICE	1081
INTRODUCTION.....	1081
VALISERVICE ENGINE.....	1081
PROCESS QUEUE LIST.....	1082
CONFIGURATION OF VALISERVICE	1085
CHAPTER 16: VALIPRODUCTION.....	1091
.SECTION 1: INTRODUCTION.....	1091
VALI SOFTWARE SUITE.....	1091
VALIPRODUCTION.....	1091
INFORMATION FLOW.....	1092
SYNERGY BETWEEN VALIPRODUCTION AND VALIPERFORMANCE	1093
VALIPRODUCTION INTEGRATION.....	1093
VALIPRODUCTION WORK-FLOW.....	1095
VALIMANAGER APPLICATION.....	1096
.SECTION 2: CONFIGURING A VALIPRODUCTION APPLICATION.....	1098
A FLEXIBLE CLIENT/SERVER SCHEME.....	1098
USER RIGHTS	1099
APPLICATION DIRECTORY STRUCTURE AND PROCESSING.....	1099
CREATING A NEW VALIPRODUCTION APPLICATION.....	1100
CONFIGURING THE WORKFLOW	1104
TRACK CHANGES.....	1111
.SECTION 3: USING VALIMANAGER.....	1112
STARTING VALIMANAGER.....	1112
RUNS OVERVIEW.....	1112
DIGGING INTO RUNS	1120
STARTING RERUN.....	1121
CHAPTER 17: VALIFORMS	1127

VALI Help

.SECTION 1: INTRODUCTION	1127
VALIFORMS OBJECTIVES	1127
USING VALIFORMS IN A VALIPRODUCTION APPLICATION.....	1130
USING VALIFORMS IN A VALIPERFORMANCE APPLICATION.....	1130
.SECTION 2: CONFIGURING VALIFORMS	1131
USING VALIFORMSBUILDER	1131
.SECTION 3: USING VALIFORMS	1178
STARTING VALIFORMS	1178
USING VALIFORMS IN INTERACTIVE MODE	1179
CHAPTER 18: THERMO	1185
.SECTION 1: INTRODUCTION	1185
.SECTION 2: USING THERMO	1187
STARTING THERMO	1187
OVERVIEW OF THERMO.....	1188
MENU BAR.....	1189
TOOLBAR.....	1190
TABS VIEW.....	1190
STATUS BAR.....	1191
OPEN MODEL FILES	1191
THERMODYNAMIC PROPERTIES CALCULATION.....	1192
CHANGING PHYSICAL UNITS.....	1195
SAVE RESULT IN THE MODEL FILE.....	1197
EXPORT TO EXCEL	1197
DRAWING LIQUID – VAPOR EQUILIBRIUM.....	1199
DRAW CUSTOMIZED CHART	1201
GRAPH TAB.....	1204
WHITE EQUILIBRIUM.....	1210
SAVING, LOADING AND COMPARING RESULTS	1212



belsim

CHAPTER 1

WHAT'S NEW

SECTION 1: USER'S GUIDE ORGANISATION

The ValiModeller user's guide is divided in 6 parts.

- **INTRODUCTION TO VALIMODELLER:** presentation of some important concepts of BELSIM environment and of data validation principles. Being familiar with these concepts is essential to be able to use ValiModeller
- **THERMODYNAMIC CALCULATION METHODS:** presentation of the various thermodynamic methods to calculate liquid-vapor equilibrium, enthalpies, etc., and list of available compounds
- **HOW TO BUILD YOUR DATA VALIDATION MODEL:** description of the different steps required to build a validation model of a process, introduction to the use of ValiModeller to define and maintain your model. Finally a practical example is used to show the major steps of a model development
- **HOW TO RUN YOUR DATA VALIDATION MODEL:** explanation of how to run ValiEngine from ValiModeller, presentation of various features like the ValiEngine runtime options, the TAG and MEA files to import and export information to and from a model, the customization of the reports and the sensitivity analysis. It gives some Tips and Tricks and lists error messages issued by ValiEngine
- **DATA VALIDATION UNITS:** detailed information on the different types of Units that can be used in a data validation model. Some practical examples are shown. Each Unit type has its own section
- **FLEX LANGUAGE DESCRIPTION:** details of the FLEX language that can be used to enter user-defined constraints and measurements

SECTION 2: NEW FEATURES IN RELEASE 4.12

This Section summarizes the new features of Release 2016 of Vali 4.

Release 4.12 is upward compatible from Release 4.11, but not backward. Files created or modified using Release 4.12 will not be readable with previous Releases.

2.1 INSTALLATION

The installation procedure uses the classical Windows Installation engine. See the installation guide (Installation Guide.pdf) for detailed installation guidelines and latest information on the software.

2.2 UPGRADING MODELS FROM RELEASE 4.11 TO RELEASE 4.12

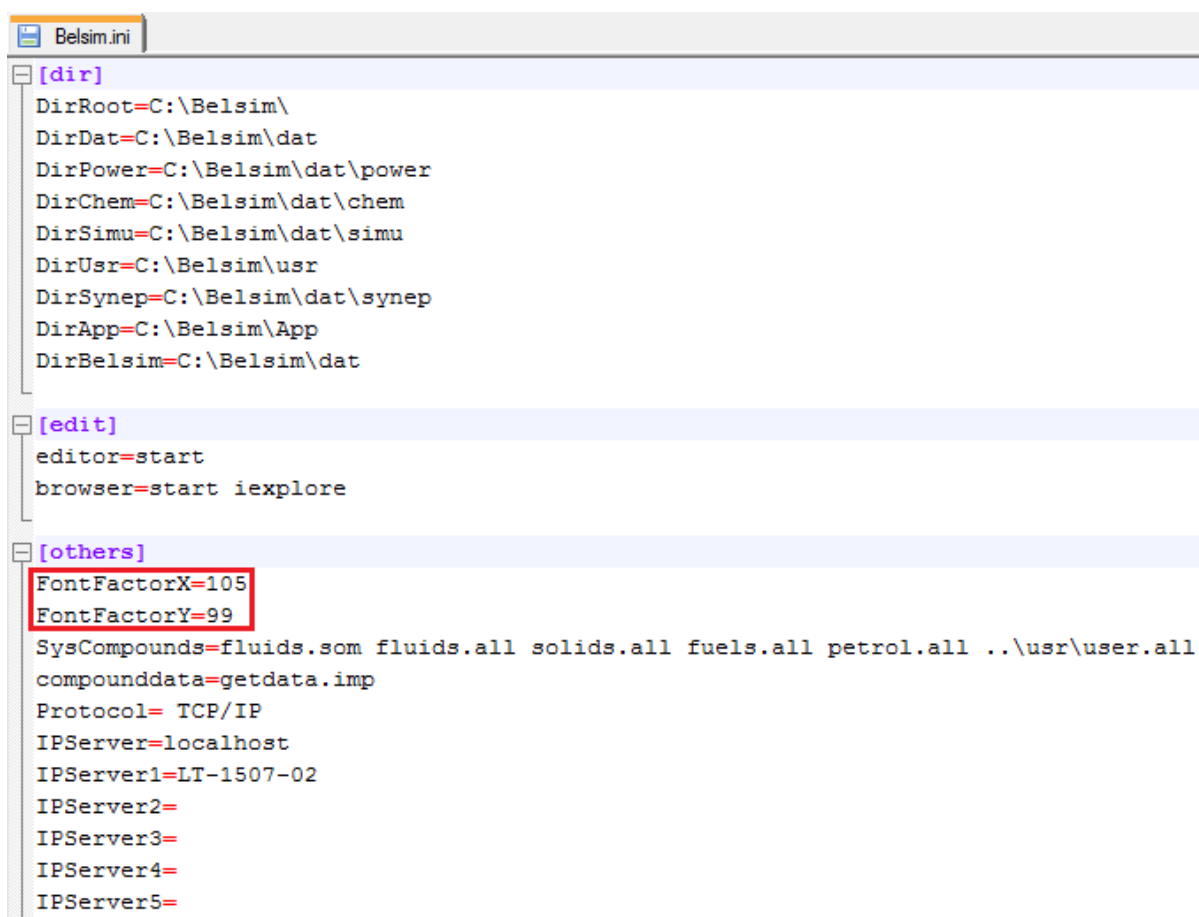
Each model should first be opened and saved with ValiModeller before being processed by a VALI engine.

2.3 KNOWN ISSUES

VALIMODELLER

You can experience some troubles using ValiModeller on Operating Systems with Asian display languages. The font size of ValiModeller tabular Notebooks can appear bigger or smaller than expected. This display issue differs depending on your Operating System version (Windows 7, 8, 10 ...).

A workaround has been developed to adapt the font size and the display. In the "**others**" section of **Belsim.ini** configuration file (located in the VALI installation folder, under "..\Belsim\dat\"), you can define and customize the **FontFactorX** and **FontFactorY** parameters (expressed in percentages). These parameters will adapt the rendering (height and width) of the tabular Notebooks.



```
[dir]
DirRoot=C:\Belsim\
DirDat=C:\Belsim\dat
DirPower=C:\Belsim\dat\power
DirChem=C:\Belsim\dat\chem
DirSimu=C:\Belsim\dat\simu
DirUsr=C:\Belsim\usr
DirSynep=C:\Belsim\dat\synep
DirApp=C:\Belsim\App
DirBelsim=C:\Belsim\dat

[edit]
editor=start
browser=start iexplore

[others]
FontFactorX=105
FontFactorY=99
SysCompounds=fluids.som fluids.all solids.all fuels.all petrol.all ..\usr\user.all
compounddata=getdata.imp
Protocol= TCP/IP
IPServer=localhost
IPServer1=LT-1507-02
IPServer2=
IPServer3=
IPServer4=
IPServer5=
```

As an example, the recommended parameters for Korean Operating Systems are:

- **FontFactorX** = 105
- **FontFactorY** = 99

VALIREPORT

Depending on the various .NET frameworks installed on your machine, you may experience some problems while starting ValiReport. Contact Belsim support if you experience some troubles.

In Office 2007, you may experience issues with the format of the date displayed by "rundate" tag in the run sheet. This can result in wrong date format when manually inserting runs in the ValiDatabase, leading to wrong date interpretation.

ValiReport is NOT compatible with MS Office 2010, 2012 and 2013 64bits. Make sure you use a 32bits version of MS Office to be able to use ValiReport.

2.4 COMPATIBILITY

VALI suite is compatible with Windows 7 and Windows 8, provided screen resolution is set to 100% for font size. If you experience display troubles in ValiModeller, check this parameter.

2.5 NEW FEATURES IN VALIMODELLER AND VALIENGINE

USER-DEFINED DIGITS NUMBER IN MAIN.ARC FILE

By default, ValiEngine prints Tag values in the Archive file with 6 significant digits. The number of decimals in the main.arc file can be defined by the user. Refer to [Output and Archive Files](#) for more details.

PRODUCT IMBALANCE FLAGGED AS ERROR

In Production Accounting applications, ValiEngine may identify product imbalances, despite the fact that we force the balance to be closed.

These product imbalances are reported as errors ("ERS") in the r1v.html log file, in addition to the r5v.html "Suspected Tags" report. Refer to [Browsing result files](#) for more details about HTML reports generated by ValiEngine.

COMPATIBILITY WITH CHINESE AND KOREAN ENVIRONMENTS

ValiModeller and ValiEngine are now compatible with Chinese and Korean environments. That means the VALI solution will behave properly in these specific environments. Nevertheless, English remains the only supported language (character set) within ValiModeller interface.

2.6 NEW FEATURES IN VALISCHEDULER AND VALISERVICE

NEGATIVE TIME DELAY IN REAL TIME APPLICATIONS

For real time applications with a long time step (e.g. weekly applications), a negative time delay parameter can be defined.

This way, the user does not have to wait a full time step (e.g. a week) to retrieve the results of the previous period.

Refer to section [Starting Real Time](#) for more details about real time applications in ValiScheduler.



belsim

CHAPTER 2

INTRODUCTION TO VALI

SECTION 1: DVR PRINCIPLES

1.1 INTRODUCTION

This Chapter presents the basic concepts you should be familiar with in order to use VALI.

Section 2	presents the principles of DVR (Data Validation and Reconciliation)
Section 3	describes VALI Environment
Section 4	introduces the objects to be used in order to build a VALI model
Section 5	shows which equations are generated by the different types of units
Section 6	deals with measurements uncertainty and redundancy. It also gives the list of measurements that can be defined
Section 7	presents some concepts specific to Petroleum Refining applications
Section 8	discusses compatibility issues with the VDI-2048

1.2 WHAT IS VALI ?

VALI is an equation-based data validation and reconciliation (DVR) software. It uses information redundancy and conservation laws to correct measurements and convert them into accurate and reliable information. VALI is used in upstream, refinery, petrochemical, chemical plants as well as power plants including nuclear power stations. VALI detects faulty sensors and pinpoints degradation of equipment performance (heat rate, compressor efficiency, etc.)

The plant measurements, including lab analyses, are reconciled in such a way that mass (on a component per component basis) and heat balances are satisfied. When necessary, L/V equilibrium and performance constraints can be added. Unmeasured values are calculated and VALI also quantifies the precision of reconciled values. Its sensitivity analysis tool shows the interdependence between the measurements. VALI can be used on-line or off-line and has been integrated in various control systems.

1.3 WHAT ARE THE BENEFITS OF DVR ?

There are a large number of benefits due to data validation and reconciliation, and they include:

- improvement of measurement layout
- decrease of number of routine analyses
- reduced frequency of sensor calibration; only faulty sensors need to be calibrated
- on-line optimization tools work with more accurate information
- systematic improvement of process data
- early detection of sensors deviation and degradation of equipment performance
- correct plant balances for accounting and performance follow-up
- quality at process level

1.4 THE BASICS OF DATA VALIDATION AND RECONCILIATION

To understand the basic principles of DVR, one must first recognise that plant measurements (including lab analyses) are not 100% error free. When using these measurements without correction to generate plant balances, one usually gets incoherence in these balances.

Some sources of errors in the balances directly depend on sensors themselves:

- intrinsic sensor precision
- sensor calibration
- sensor location

A second source of error when calculating plant balances is the small instabilities of the plant operation and the fact that samples and measurements are not exactly taken at the same time. Using time averages for plant data partly reduces this problem. However, lab analyses cannot be averaged.

Finally, one must also realise that in some cases too many measurements are available whereas in some other cases some measurements are missing and must be back calculated from other measurements. The aspect of data redundancy is here an essential factor. It is because too many measurements are available that one can prove the fact that the measurements are somewhat inaccurate. For example, measuring all inlet and outlet flow rates of a plant (or of part of a plant), usually leads to some imbalance between the total inputs and total outputs.

The main idea of DVR is to use the data redundancy of the system as the source of information to correct the measurements. In fact, each measurement is corrected as slightly as possible but in such a way that the corrected measurements matches all the constraints (or balances) of the process.

Mathematically, this can be expressed as:

$$\text{Min } \sum_i \left(\frac{y_i^* - y_i}{\sigma_i} \right)^2$$

subject to $F(x_j, y_i^*) = 0$

$$G(x_j, y_i^*) \geq 0$$

where : y_i^* is the validated value of measurement i

y_i is the measured value of measurement i

x_j is an unmeasured variable j

σ_i is the measurement uncertainty or confidence interval

$F(x_j, y_i^*) = 0$ corresponds to the process equality constraints

$G(x_j, y_i^*) \geq 0$ corresponds to the process inequality constraints

The term $\left(\frac{y_i^* - y_i}{\sigma_i} \right)^2$ is called the penalty of measurement i

SECTION 2: VALIMODELLER ENVIRONMENT

2.1 VALIMODELLER COMPONENTS

ValiModeller is a multi-purpose modelling tool intended to develop and maintain one or several process models. ValiModeller can be used for offline validation and engineering studies.

It contains the following modules:

- Vali.exe: ValiEngine, which will validate and reconcile the measurements of your process
- ValiModeller.exe: the graphical user interface GUI of VALI; allows you to create and maintain your DVR models
- Iofile.exe: a stand alone application that can be used to process measurement files (and reports) independently of ValiModeller
- Sensiv.exe: the sensitivity analysis tool of ValiModeller and analyses the interdependencies between measurements and process variables

2.2 THE MODEL FILE (BLS) AND OTHER FILES

ValiEngine and ValiModeller GUI work with the information stored in a file called the **Model file** (e.g. model.bls). This file contains the description of the models as well as their data and their results.

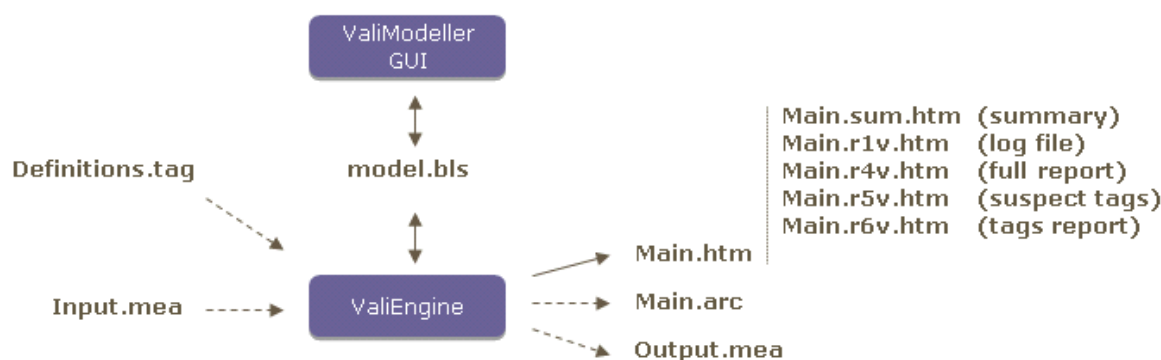


FIGURE 1 - VALI ENVIRONMENT

ValiModeller allows you to develop your DVR model and to visualise, after a run, the state of the process by looking at the units and streams data. The data are read from the model file and are written back in it as well.

The measurements can be defined either within the model file, using ValiModeller, or in one or several TAG files, using for example NotePad or WordPad text editor. A TAG file is a simple ASCII file where each line defines one measurement by giving it a name, indicating the variable it relates to, and defining a default value and precision.

Example:

```
TI-001 Stream Feed    T      120    5      C
```

You can also have MEA files that give, line by line, the current values of the measurements. Typically in an on-line environment, these files would come from the process control computer and from the LIMS (Laboratory Information Management System).

Example:

```
TI-001 98      C
```

The MEA files can also be used to modify the model, like changing various Units parameters or specifying that parts of a model must be ignored or are currently shutdown.

Finally, MEA files can be used to generate dedicated reports or transfer information to the outside world for archiving, reporting or integrating in other applications.

VALI reads the model description from the model (bls file) and the measurements definitions and values from the TAG and MEA files. It creates a number of report files (r1v, r2v,..., r6v) with various messages and results and optionally output MEA files with validated values of selected measurements.

In addition, when the run is successful, all the model data are saved in the model file.

VALI can also generate ARC files with all data related to tags (among which measured, filtered, validated values,) as well as some information on the run (total penalty, penalty by PFD, number of filtered and eliminated tags, etc.). This file can be loaded into ValiDatabase through ValiDBLoader application.

ValiEngine is described in a section "[How to run your data validation model](#)".

SECTION 3: VALI OBJECTS

Within the VALI graphical user interface (GUI), various objects are used to describe the process and problem.

The objects available are:

- PFD: sketch of the process, graphically showing the units and their interconnecting streams. When needed, complex processes can be divided in several different PFDS
- UNIT: a piece of equipment of the process, like a pump, a heat exchanger or a reactor
- STREAM: connection between units
- THERMOD: list of chemical compounds and thermodynamic methods. Each material STREAM has an associated THERMOD indicating to VALI what compounds can be present in this material STREAM and what methods must be used to compute its thermodynamic properties
- REACTION: chemical reaction defined by its stoichiometry
- COMPOUND: chemical compound, like H₂O or BENZENE with its thermodynamic properties
- TAG: measurement point associated to a model variable, like a stream temperature, and by assigning a name and an accuracy to it

Each object must have a name of 32 characters maximum. These names must be unique within each class. UNITS and PFDS are however considered as being of the same class. A UNIT and a PFD can therefore not bear the same name.

For COMPOUNDS, only the first 30 letters are significant and should be unique, although 32 letters are defined and kept.

When a tagname is automatically built by ValiEngine, its length may be up to 48 characters.

For example, a movement with submovement tagname may be written in the *.arc file as:

ZONE_PRODUCT_MOVEMENT_SUBMOVEMENT_MEASUREMENT,

each part being maximum 8 characters long + 4 underscores = 44 maxi, rounded to next multiple of 8.

3.1 PFD

Process Flow Diagrams are organised recursively, in a very similar way as directories under Windows operating system: any PFD can contain other PFDs. The **main PFD**, called by default MAIN, can only contain other PFDs, but no UNITS (process equipment). The other PFDs can contain:

- PFDs
- UNITS, which describe the pieces of equipment of the process
- STREAMS which connect the UNITS

For coherency, any UNIT and any PFD can appear in only one PFD. Similarly, any stream can connect a maximum of two different UNITS. However, these two UNITS can be located in two different PFDs.

When running VALI you will indicate what PFD must be validated. All units and sub-PFDs of this PFD will then be part of the model.

3.2 UNIT

The various types of UNITS are gathered into categories: heat exchangers, reactors, etc. For each category, one or several different types are available with various capabilities. For example, a heat exchanger can be modelled by different types of units like EEXVAL, HEXVAL, SATVAL, WALVAL (see "[Process unit equations](#)" for more information on the various types of units).

UNIT DATA defines the operation of the UNIT. In addition to a MODE parameter, which defines the type of balances to perform, several other parameters further describe the current operation of the unit: pressure drops, extents of reactions, etc.

3.3 STREAM

Objects called Streams represent the connections between process Units. They describe what flows from one equipment to another.

There are three types of streams:

- MATERIAL: a stream of chemical compounds
- THERMAL: heat flow
- MECHANICAL: mechanical power transfer

MATERIAL STREAMS

A MATERIAL STREAM gathers all the information on a particular stream: pressure, temperature, enthalpy, rate and composition. Each STREAM has a state indicator, that determines the phases that can be present:

- Liquid: liquid only
- Vapour: vapour only
- Liquid/Vapour: liquid and/or vapour
- Liquid/Liquid: up to two liquid phases can appear
- Water/Liquid: one water phase (assumed to be pure) and one liquid phase can appear
- Water/Liquid/Vapour: a water, a liquid and a vapour phase can occur
- Liquid/Liquid/Vapour: two liquid and one vapour phases can appear

For example, a stream indicated as LV can be in liquid state, in vapour state or can be a mixture of these two phases. However, only one of these two phases can happen to be present.

On the contrary, setting a stream as L will force VALI to consider the stream as a liquid, even when its validated composition, pressure and temperature would correspond to a vapour state.

Attention: for STREAMS where the saturation state is specified, like the outlet streams of a liquid/vapour separator, it is recommended to set the states of the streams accordingly: liquid for the saturated liquid and vapour for the saturated vapour. This significantly improves the convergence of the resulting model.

THERMAL AND MECHANICAL STREAMS

The heat flow supplied to a unit is normally defined as an inlet THERMAL STREAM, while the heat produced by a unit should be defined as an outlet THERMAL STREAM. However, you do not imperatively have to follow up this rule. Negative values would then be obtained for the heat load.

The same rule applies for MECHANICAL STREAMS.

3.4 THERMOD

The VALI object THERMOD represents the set of information required to calculate the thermodynamic properties of a STREAM:

- the names of the Compounds which can appear in the STREAM
- the names of the methods to use to calculate its thermodynamic properties
- binary interaction parameters (if necessary)

3.5 REACTION

The object called REACTION contains the information related to a chemical reaction:

- the COMPOUNDS involved in the reaction
- their stoichiometric coefficients

3.6 COMPOUND

A COMPOUND object identifies a chemical component involved in the process and contains all its basic properties. There are three types of compounds:

- EPIC: chemical compound defined by its formula. The compound must be in the liquid and/or vapour phase
- SOLID: chemical compound defined by its formula. The compound must be in the solid phase
- PETFRA: petroleum fraction that is in the liquid and/or vapour phase. The chemical formula and properties of the compound are estimated from its density and its boiling temperature

VALI handles compounds of type PETFRA exactly as usual EPIC compounds. The only difference is the way their properties are generated.

On the contrary, SOLID compounds are handled differently. They constitute a distinct phase and cannot melt or vaporise and become a liquid or a vapour. In separators, SOLIDS can be entrained with the gas phase and/or with the liquid phase. Solids can be involved in chemical reactions as other compounds.

Attention: it is possible to model phase changes to and from the solid state by defining a (pseudo) chemical reaction, where a SOLID compound is transformed into its corresponding liquid/gas compound, with the same molecular formula.

3.7 TAG

TAGs define the properties of measurements.

The definition of a measurement contains its location (on a STREAM or a UNIT), its name, its value and its accuracy. Optionally, it is possible to associate to each tag:

- bounds: warning, error and solver
- filter settings (ValiFilter licence required)
- compensation settings
- control strings, to generate warning messages and to update the measured value or its precision, depending on the measured value

In a model, four sets of values and precisions are attached to each tag:

- the default value and its precision, which are the values as defined in the model (or in the tag file) by the user
- the measured value and its precision, which are initially set to the default values but can be overwritten by values coming from measurement files (MEA files) by ValiEngine
- the used value and its precision, which are the values actually used during the reconciliation step. They are initialized to the measured values but can be overwritten by filters, compensation algorithms or control strings
- the reconciled value and its precision as validated by ValiEngine

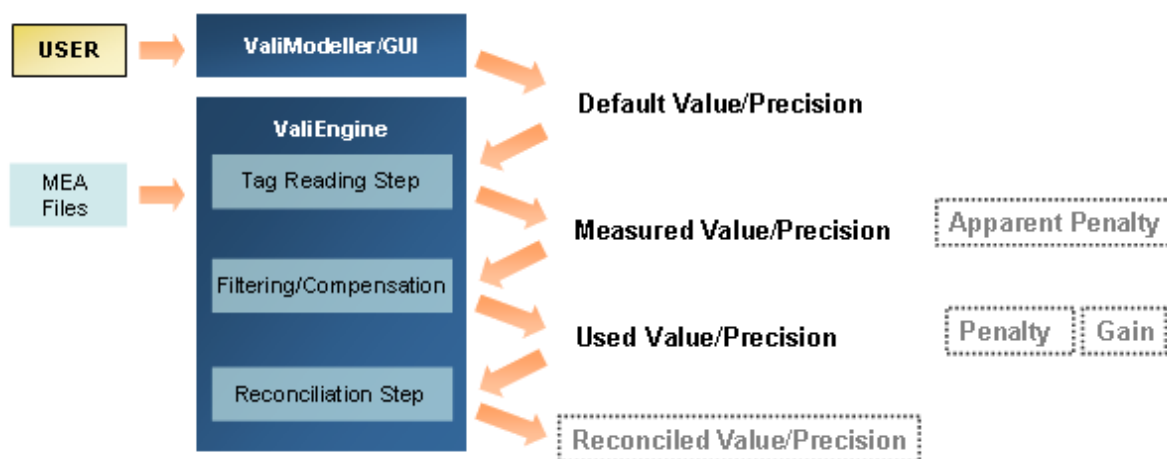


FIGURE 2 - THE TAG VALUES

In addition, ValiEngine will calculate for each tag:

- the apparent penalty: calculated on the basis of the measured value:

$$\text{ApparentPenalty} = \left(\frac{\text{reconciled}_i - \text{measured}_i}{\sigma_{i,\text{measured}}} \right)^2$$

- the penalty: calculated on the basis of the used value, can be selected (through the mathematical unit OPTVAL) as being either the measurement penalty, the correction penalty or the measurement impact:

$$\text{MeasurementPenalty} = \left(\frac{\text{reconciled}_i - \text{used}_i}{\sigma_{i,\text{used}}} \right)^2$$

$$\text{CorrectionPenalty} = \left(\frac{\text{reconciled}_i - \text{used}_i}{\sigma_{i,\text{correction}}} \right)^2$$

Impact = decrease in total penalty when setting the measurement OFF

- the gain: which expresses the accuracy improvement and is calculated on the basis of the reconciled and the used accuracies:

$$\text{Gain} = 100 \cdot \left(\frac{\sigma_{i,\text{used}} - \sigma_{i,\text{reconciled}}}{\sigma_{i,\text{used}}} \right)$$

3.8 AN EXAMPLE: THE LIQUID-VAPOUR SEPARATOR

Figure below represents the VALI objects that describe a liquid-vapour separator.

The process unit consists in a single UNIT named D. The three streams connected to the separator are S1, S2 and S3. A single THERMOD named TMOD, attached to each STREAM, represents the set of information necessary to calculate their thermodynamic properties. The compounds, listed in the THERMOD, describe the chemical compounds CHEM1, CHEM2 and CHEM3.

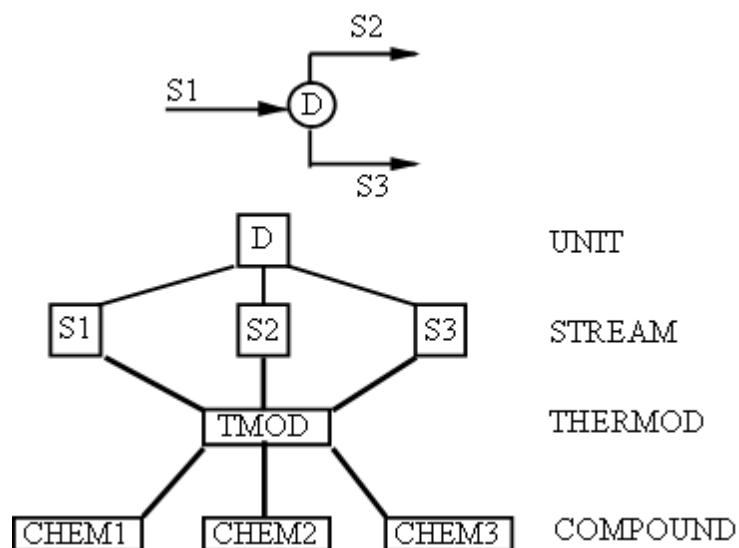


FIGURE 3 - THE LIQUID-VAPOUR SEPARATOR

SECTION 4: PROCESS UNIT EQUATIONS

The set of equations used to validate the measurements is built up on basis of equations subsets, written by VALI for each unit of the process. Among these process units you may find:

- black boxes
- splitters
- mixers
- heat exchangers
- compressors
- turbines
- pumps
- chemical reactors
- separators (liquid/vapour, liquid/liquid, etc.)
- distillation, absorption and stripping columns
- controllers
- storage tanks
- valves

Each unit used to model these pieces of equipment will generate a number of equations:

- the mass balance for each chemical compound
- the energy balance of the unit
- phase equilibrium equations
- performance equations
- pressure drop equations

When validating composition and flow rate measurements, only mass balances are necessary.

When temperature measurements are available, you can use energy balance equations. This will not only validate these measurements but can also improve the accuracy of the compositions and flow rates measurements.

Moreover, when streams are involved in a phase change phenomenon, as for example in liquid/vapour separator, the liquid-vapour equilibrium relationships or the saturation equations can help to improve the accuracy of the composition and pressure measurements.

For some units, it is also possible to consider performance equations. This allows you, for example, to avoid temperature crossings in heat exchangers or non-feasible efficiencies in rotating equipment.

According to the type of equipment you want to model, one or several VALI unit types are available, with various capabilities. Each UNIT type has a six to eight-letter name, which usually ends by VAL and where the first characters indicate the type of unit, like BBXVAL for a Black box.

The first paragraph hereafter explains the way to control what equations are generated by each unit. The following paragraphs explain how to select the appropriate VALI UNIT type in different situations. They also describe what equations are generated.

4.1 CONTROLLING THE SET OF GENERATED EQUATIONS

The equations that are generated obviously depend on the type of UNIT you will select. In addition, the UNITS have various parameters that allow you to further define the equations generated:

- the MODE parameter controls energy, phase equilibrium and performance equations
- the MODEDP and REFDP parameters controls pressure drop equations
- specific unit parameters are sometimes available

The set of equations can also change depending on the status of streams and units.

A UNIT (or a complete flow-sheet) can be specified as OUT. The model is handled as if the UNIT (or PFD) had never been defined.

A STREAM can be set OFF, meaning that no flow rate is present. All measurements on that stream are deactivated, its variables are removed from the equations system and its rate is set to zero. Similarly, a UNIT (or even a complete PFD) can be set OFF. All the STREAMS connected to the UNIT are then set OFF.

A propagation algorithm checks whether the fact that a STREAM is off does automatically entrain that a UNIT should be set OFF as well. For example if the inlet material STREAM of a pump is set OFF, then the pump must be set OFF as well. See sub-section "[ON/OFF propagation](#)" for more information.

MASS BALANCES AND STREAM RATES SHARING

In order to reduce the number of variables and equations, VALI uses a feature called **Stream Rates Sharing** allowing streams having the same composition to share the same partial rates variables. This unique feature is totally automatic and transparent to the user. It can however be useful to understand how this feature works, given that it affects the way units generate their equations.

For example, in a pump, the inlet and outlet streams obviously have the same composition. Rather than duplicating the partial rates of the inlet stream for the outlet stream, VALI will instead use the rates of the inlet stream for the outlet stream. If the inlet stream involves 10 compounds, this avoids generating 10 extra variables and extra equations.

This concept is further generalized to the case where a stream is split up in several streams of the same composition. In this case, an additional variable, the rates (mixture) fraction (MIXF) is generated for each of the outlet streams. The actual flow rates of the outlet streams are obtained by the product of their rates fraction by the flow rate of the split stream.

VOLUME BALANCES

When dealing with global material balances of plants without any tracking of the stream composition, volume balances can be added to the mass balances of some of the units. The advantage of using volume balances is that this increases the redundancy of the model and thus improves the accuracy and robustness of the validated values. Of course, volume balances should only be used for the units where volume will be conserved, like most blending units or cold storage tanks.

To activate volume balances, you must introduce the keyword **CONVDENS** in the unit **COMMENT** and provide density measurements (**CONVDENS** measurements) on the streams. This feature only works on units of type **BBXVAL**, **NETVAL**, **COLVAL** and **TNKVAL**.

Example:

see model `convdens.bls` in the directory "`\examples\convdens`".

Attention: Volume balances should not be used when dealing with thermodynamic-based models, that is models including a detailed representation of the chemical compounds. Indeed the density of the streams is in this case linked to the stream composition and thus adding volume balances would artificially and incorrectly increase the redundancy of the model.

IMBALANCES

In ValiProduction applications, it is sometimes necessary to act an imbalance, for example when an unidentified loss has occurred. The corresponding imbalance can be acted on a storage tank (See unit **TNKVAL**) or on a product (see unit **NETVAL**).

USING THE PARAMETER MODE

Most units have a parameter called **MODE** that enables you to control whether energy, phase equilibrium or performance equations must be generated by the unit or not. Note that mass balances are always generated.

Depending on the Unit type, different **MODE** values can be defined, like for example:

- **MODE=0**: all unit equations are generated
- **MODE=1**: only mass balances are generated
- **MODE=2**: only energy and mass balances are generated
- **MODE=3**: only phase equilibrium and mass balances are generated

USING PARAMETERS **MODEDP** AND **REFDP**

To control whether pressure drop equations are generated, you use the parameters **MODEDP**.

Depending on the Unit type, different **MODEDP** values can be defined, like for example:

- MODEDP=0: no pressure drop equation will be generated
- MODEDP=1: a pressure drop equation will be generated
- MODEDP=2: a pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * (Massf / MassfX)^2$$

where :

- CFDP is a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20% to 30%)
- DPX is the nominal pressure drop
- MassfX is the nominal flow rate
- Massf is the actual flow rate

Attention:

1. Units accepting only one outlet material stream have only one set of parameters DP, CFDP, etc. whereas units which can handle several outlet streams have several pressure drop parameters called DP1, DP2, ..., CFDP1, CFDP2, etc.
2. To simplify the modelling it is possible to specify (see parameter PRESMODE of unit OPTVAL) that pressure and/or pressure drops will be considered as constant when not measured. This feature must not be used when using equations relating pressure drops and rates.

When a unit mixes several inlet streams, you can control whether the inlet streams pressures are related or not with the REFDP parameter, which can be:

- 'All Inlet': all inlet streams have the same pressure
- <stream>: inlet stream pressures are not related. If pressure drops equations are generated, then the stream indicated by REFDP is taken as reference for pressure drop calculations

By default REFDP points to the first inlet stream.

If the stream referred by the REFDP parameters is OFF, then the next stream in the list is taken as reference and an error message is generated.

OTHER EQUATIONS

Some units can generate other types of equations, like saturation state or performance equations. Specific parameters are then available.

ELECTRO-NEUTRALITY

When dealing with ions, you can add on selected streams the constraint that the total electric charge of positive and negative ions must be in balance.

$$\sum_i (X_i \cdot \text{Electro}_i) = 0$$

Where :

- X_i is the partial molar flow rates of compound (ion) i
- Electro_i is the ionic charge of compound (ion) i

The electric charges of the compounds (ELECTRO) and the streams for which the constraint must be generated are defined through measurement files (see " [Electro-Neutrality Constraints](#)").

4.2 BLACK BOXES AND MIXERS: BBXVAL, NETVAL, EFFVAL, CUTVAL

Four different types of UNITS are available:

- **BBXVAL**: general black-box, allowing to mix a number of inlet streams and split their contents in a number of outlet streams of varying pressure, temperature and composition. THERMAL and MECHANICAL STREAMS can also be specified as inlet and outlet streams of a BBXVAL
- **NETVAL**: dedicated to the modeling of complex networks of product lines in overall mass balance models
- **EFFVAL**: mixing of a number of thermal or mechanical streams in two outlet streams: a main outlet and losses
- **CUTVAL**: to break the material or pressure loops that can appear when dealing with closed cycles (e.g. water cycle in power plant)

BBXVAL

A BBXVAL unit ensures the global mass and energy balances to be satisfied, allowing you to model mixers, distillation, absorption or stripping columns and feed drums, but sometimes, also entire parts of a process. No reactions can occur in a BBXVAL.

A BBXVAL unit can also represent shafts linking compressors, turbines and pumps.

Depending on the unit parameters, the unit generates:

- a mass balance equation for each chemical compound
- an energy balance around the black box. If vaporisation or condensation occurs, the heat involved in this phase change is, of course, taken into account. When inlet and outlet pressures are not equal, the energy balance takes the expansion and compression works into account
- equalities between the temperature of all outlet streams
- equality between the temperature of all outlet streams and the first inlet stream
- pressure drop equations

NETVAL

The NETVAL unit has been primarily designed to simplify the modelling of tank farms where the storage tanks can be used to store different types of products in function of time.

Rather than having to change the structure of the model by reconnecting streams from and to the different tanks, the streams are given an ATTRIBUTE that defines their product type. The streams coming from or going to the process units and the tanks are all connected to units of type NETVAL, all units of this type being considered as part of a single global network of pipes. The NETVAL units will generate as many balances as there are attributes and for each of them will include all the streams of the corresponding attribute.

Attention: NETVAL units can only be used in mass (or mass and volume) balance mode.

EFFVAL

An EFFVAL unit generates the heat balance between all inlet streams and the two outlet streams. Any combination of thermal and mechanical streams can be used for the inlet and outlet streams. The yield of the unit can be expressed as a loss fraction or as efficiency.

This unit is particularly suited for modelling motors, generators or turbine shafts.

CUTVAL

The type CUTVAL is a "black-box" UNIT ensuring energy and optionally pressure balances but no mass balance. It is useful in close loops where the mass balances can create singular matrix.

SPLITTERS: SPLVAL

A Splitter splits a stream in up to ten outlet streams having the same composition.

Depending on the unit parameters, the unit generates:

- using the Stream Rates Sharing concept, a global mass balance equation linking the Rates fractions of the outlet streams to that of the inlet stream
- one equation for each outlet stream, to determine the split fraction versus the inlet stream
- an energy balance
- one pressure drop equation. With the parameter DP, you can specify (or measure) the pressure drop between the inlet stream and all outlet streams

Attention:

1. The outlet streams of a SPLVAL are always set at the same pressure, even if a pressure drop is not specified.

2. All streams must be set in the same State (L,V,LV, etc.).

4.3 HEAT-EXCHANGERS: EEXVAL, HEXVAL, PRHVAL, SATVAL, WALVAL, ECOVAL

You can model heat exchangers by using six different types of units:

- **EEXVAL:** a very flexible model that can have from 1 up to 10 inlet and outlet material streams as well as thermal streams. Any heat exchange can occur provided that the global heat balance of the exchanger is satisfied
- **HEXVAL:** dedicated to co- and counter-current exchangers, with 2 inlet and 2 outlet material streams, and takes a heat exchange surface area and transfer coefficient (convection and/or radiation) into account. Saturation constraints can be added to any outlet stream (saturated liquid or vapour). The heat transfer coefficient and the pressure drops can be related to the flow rates
- **PRHVAL:** preheaters where a stream is heated up by a vapour that is desuperheated, condensed and subcooled. An optional drain inlet allows connecting a third (condensed) inlet stream
- **SATVAL:** set the vapour fraction of one outlet stream
- **WALVAL:** estimating heat losses through the walls of vessels
- **ECOVAL:** accurately model HRSG (Heat Recovery Steam generators) elements

EEXVAL

Depending on the unit parameters, the unit generates:

- global mass balance equations linking the Rates fractions of the outlet streams to that of the corresponding inlet streams, using the Stream Rates Sharing concept
- an overall energy balance. If condensation or vaporization occurs, the enthalpy of the phase change is taken into account
- one pressure drop equation per outlet material stream. With the parameter DPi, you can specify (or measure) the pressure drop between the corresponding inlet and outlet streams

HEXVAL

The **HEXVAL** unit type allows you to consider the heat exchanger performance and to avoid temperature crossings in the exchanger. Contrary to the **EEXVAL** type, an **HEXVAL** can only handle heat exchange between 2 pairs of inlet/outlet streams. In addition to the heat and mass balances, this unit type can generate a performance equation linking the exchanger load to the temperature difference between the hot and cold streams.

The default method is based on DTLN concept:

$$\text{LOAD} = A \cdot U_{\text{glob}} \cdot \text{CORDTLN} \cdot \text{LNTD} + G \cdot A \cdot \sigma \cdot (T_{\text{hot}}^4 - T_{\text{cold}}^4)$$

$$\text{LNTD} = \frac{\text{DTIDE1} - \text{DTSIDE2}}{\ln(\text{DTSIDE1} / \text{DTSIDE2})}$$

Where :

- DTSIDE1 and DTSIDE2 are the temperature differences on each side of the heat exchanger
- σ is the constant of Boltzmann ($5.667 \cdot 10^{-8} \text{ W/m}^2\text{K}^4$)
- Thot is the outlet temperature of the hot stream
- Tcold is the outlet temperature of the cold stream
- Uglob is the heat transfer coefficient
- A is the heat transfer surface area
- CORDTLN is a correction factor that accounts for the deviation from true counter-configuration (only for multi-passes exchangers)

Alternatively, a method based on the concept of Number of Transfer Units (NTU) can also be used to model convective heat transfer. This method is recommended when the difference of temperature between the hot and cold streams is small on one or both sides of the exchanger.

The NTU method is based on the notion of efficiency, the efficiency being defined by the ratio between the actual heat load of the exchanger and the maximum heat load achievable assuming an infinite heat transfer surface area.

$$\text{EFF} = \text{LOAD} / \text{LOADMAX}$$

The efficiency is calculated on basis of the number of transfer unit NUT according to various formulas (see description of unit [HEXVAL](#)) depending on the exchanger configuration and operation mode.

$$EFF = F(NUT, C)$$

$$NUT = U_{glob} * A / C_{min}$$

$$C = C_{min} / C_{max}$$

Where :

- C_{max} and C_{min} are the total heat capacities of the streams ($Massf * C_p$)

With both methods (DTLN or NTU), you can define measurements on temperature differences (DTSIDE1, DTSIDE2), heat load (LOAD), heat exchange area (A) and heat exchange coefficients (by convection U_{glob} and by radiation G). Among these three latest variables (A, U_{glob} and G), at least two of them must be, either measured, or set constant.

Alternatively the heat transfer coefficient U_{glob} can be calculated in function of the flow rates of the streams according to the following formula:

$$1/U_{glob} = FF + 1/(CFH.U_{clean})$$

Where :

- FF is a fouling factor
- CFH is a tuning parameter
- U_{clean} is the clean heat transfer coefficient calculated from:

$$1/U_{clean} = (1/U_1 + RTUB + 1/U_2)$$

Where :

- U_1 is the film heat transfer coefficient on shell side
- RTUB is the tubes thermal resistance
- U_2 is the film heat transfer coefficient on tubes side

U_1 and U_2 can either be introduced as pseudo-measurements or can be calculated from reference conditions:

$$U_i = ReferU_i * (Massf_i / Massf_{U_i})^{0.7}$$

Where :

- Refer U_i is the film heat transfer coefficient at nominal conditions
- Mass f_i is the actual flow rate
- Mass f_{U_i} is the flowrate at nominal conditions

The HEXVAL unit type also allows you to impose a saturation state (saturated liquid or vapour) of

the outlet streams with the VFRACi parameter.

PRHVAL

The PRHVAL unit type is similar to a HEXVAL unit but allows you to split up the heat exchange into three regions (desuperheating, condensing and subcooling) with different heat transfer coefficients.

Note that radiation is not taken into account.

In addition to the heat and mass balances, this unit type generates three performance equations (one for each zone):

$$LOAD = A.U.LNTD$$

$$LNTD = \frac{DTIDE1 - DTSIDE2}{\ln(DTSIDE1 / DTSIDE2)}$$

Where:

- DTSIDE1 and DTSIDE2 are the temperature differences on each side of the three regions

SATVAL

This unit allows you to set the saturation state of the outlet stream: any vapour fraction between 0 and 1 can be set. It is also possible to specify vapour super-heating or liquid sub-cooling by defining the difference between the saturation state and the stream temperature.

WALVAL

This unit allows estimating the heat losses to the ambient air in function of the wall temperature and surface area. Different equations are used for vertical and horizontal walls.

ECOVAL

This type of UNIT is dedicated to the detail modelling of tube layers in heat recovery steam boilers (HRSG). It includes predictions of heat transfer coefficients and pressure drop on fumes and steam side. The same UNIT type can be used for the modelling of economisers, vaporisers and super-heaters.

Attention: ECOVAL unit is not part of the VALI standard installation. It is sold as an add-on to VALI under a separate license agreement.

4.4 COMPRESSORS AND TURBINES: COMVAL AND TURVAL

The mass balances of compressors and turbines are generated, by using the stream rates sharing algorithm. A mechanical stream must be connected as inlet of the compressor or outlet of the turbine.

In addition to the mass balance, two types of equations are generated: the heat balance and the

efficiency equations involving the isentropic or polytropic efficiencies.

The isentropic or the polytropic efficiency parameter can be introduced as a pseudo-measurement.

When the pressure balance equation is activated, the pressure ratio of the machine is computed as well.

For turbines a Stodola parameter can be defined in order to link the flow rate going through the unit to the driving pressure drop.

4.5 PUMPS: PUMVAL

The mass balance of a pump is generated by using the stream rates sharing algorithm. A mechanical stream must be connected as inlet of the pump.

In addition to the mass balance, two equations are generated: the heat balance and the volumetric efficiency equation involving the efficiency of the pump.

In the latter case, the volumetric efficiency parameter can be introduced as a pseudo-measurement.

When the pressure balance equation is activated, the pressure ratio of the machine is computed as well.

Eventually, performance curves can also be introduced:

- efficiency curve versus volumetric flow rate
- pump head versus volumetric flow rate

The distance from the performance curves is given as measurable parameters.

4.6 CHEMICAL REACTORS: BATVAL, BWHVAL, RCTVAL, RLVVAL, RKPVAL, RKIVAL, CND-HNO₃, HEX-HNO₃, RCT-HNO₃

Different types of reactors are available:

- BATVAL: any reaction can occur. The unit generates atomic balances
- BWHVAL: any reaction can occur. The unit generates atomic balances and an equilibrium equation based on the minimisation of Gibbs Energy
- RCTVAL: only the selected reactions can occur
- RLVVAL: only the selected reactions can occur. The unit has two outlet streams: one in gas phase and one in liquid phase. These streams are in physical (liquid/vapour) equilibrium
- RKPVAL : only the selected reactions can occur. Some of these reactions can be specified as chemically equilibrated
- RKIVAL: only the selected reactions can occur. The conversions are calculated on basis of a user-defined kinetic model
- CND-HNO₃: specific model for modelling nitric acid condensers
- HEX-HNO₃: specific model for modelling nitrous gas coolers
- RCT-HNO₃: specific model for modelling ammonia oxidation

These units will generate:

- mass balances in accordance with their reaction scheme
- an energy balance
- optionally, depending on the MODEDP parameter, a pressure drop equation
- for the RLVVAL unit, physical equilibrium equations between the vapour and liquid outlet streams
- for the BWHVAL and RKPVAL units, chemical equilibrium equations
- for the RKIVAL, conversion calculations based on the kinetics
- for CND-HNO₃ and HEX-HNO₃, the Bodenstein kinetics are used
- for RCT-HNO₃, the oxidation to NO and N₂ is adjusted via a Selectivity parameter

Attention: When the energy and/or the equilibrium equations of a RLVVAL unit are activated, this unit also generates equalities between outlet streams pressures and temperatures.

4.7 SEPARATORS: LVEVAL, LLEVAL, LLVVAL, WLEVAL, WLVVAL

Five different types of units allow you to handle physical separation of up to two liquid and one vapour phases:

- LVEVAL: Liquid/Vapour separator
- LLEVAL: Liquid/Liquid separator
- LLVVAL: Liquid/Liquid/Vapour separator
- WLEVAL: Water/Liquid separator
- WLVVAL: Water/Liquid/Vapour separator

According to the unit parameters, the unit generates:

- one mass balance equation for each chemical compound
- an energy balance
- one (or two) equilibrium equation for each component present in the outlet streams
- optionally, one pressure drop equation. With the parameter DP, you can specify (or measure) the pressure drop between the inlet stream and all outlet streams
- when the energy or the equilibrium equations are activated, these units also generate equalities between outlet streams pressures and temperatures

Attention:

1. If some of the partial flow rates or compositions are measured as zero with a standard deviation equal to zero in one of the outlet streams, the related liquid-vapour equilibrium relationship is not considered by VALI. This can be useful when some chemicals are practically non-existent in the liquid or vapour phase.
2. The outlet streams must be in a state compatible with their function. For example, a vapour outlet stream should be set as V.
3. No assumption is made about the destination of solids that would be present in the inlet streams. Composition measurements (or information coming from the downstream balances) should define whether solids are going with the liquid or vapour phase.
4. In Water/Liquid and Water/Liquid/Vapour, the water is assumed to be pure. The solubility of water in the organic phase is based on the solubility in kerosene.

4.8 DISTILLATION, ABSORPTION AND STRIPPING COLUMNS: COLVAL, DISVAL, CRUVAL

Distillation columns can be modelled by using units of type BBXVAL (see Black boxes), COLVAL, DISVAL or CRUVAL:

- COLVAL: generates the same equations as a BXXVAL, but in addition, it allows you to specify a saturation state (saturated liquid or vapour) for each outlet stream
- DISVAL: generates equilibrium equations for a number of theoretical trays. Distance from equilibrium can be set through a Murphree efficiency
- CRUVAL : is dedicated to refining applications. It allows identifying pseudo-compounds used to model petroleum cuts

4.9 STORAGE TANKS: TNKVAL

Storage tanks can be modelled using units of type TNKVAL. Only the material balance of the tank is taken into account. The heat balance is not taken into account.

Typically, storage tanks models are most often used in daily production balance models. In such a case, the recommended approach is to use as initial inventory (measured as constants) the validated inventory of the preceding period (usually day).

Different modelling options are available enabling to apprehend various situations and operation modes:

- the inventories can be expressed on mass, volume or level (using a strapping table) basis
- the composition of the tank can be made variable or kept constant over the validation time period
- a temporary imbalance can be activated to act a product loss or surplus

4.10 USER DEFINED EQUATIONS AND CONTROLLERS: OPXVAL, FLXVAL, VARVAL, EQUVAL

Three units are available to enter your own equations and measurements: OPXVAL, FLXVAL and VARVAL. In addition, you can add Flex code to generate additional constraints on all equipment units.

An OPXVAL unit allows you to define up to 9 simple equations involving two variables:

$$\text{VAR}_i = X_1 \# X_{i+1}$$

Where:

- X_1 and X_{i+1} are any STREAM or UNIT variables, referenced through tags
- the symbol # is one of the four operators: + - * /
- VAR_i is one of the parameters of the OPXVAL unit and is dimensionless

For example, selecting as X variables the temperatures of two streams and - as the operator, you can calculate the temperature difference between two streams. By defining a measurement on, the VAR_i parameter, you can measure this temperature difference.

Whenever you want to consider more sophisticated relationships, you can use units of type FLXVAL

or add a Flex code to any equipment units. This allows you to write equations with a FORTRAN-like programming language. Access to the thermodynamic data bank is enabled by dedicated procedures.

Virtually, any relationship can be added. For example, the relationship between temperature, pressure, composition and viscosity would allow you to enter viscosity measurements.

The VARVAL unit provides up to nine variables which can be referenced in any Flex code (on the VARVAL unit itself or on any other unit).

Attention: All variables of units OPXVAL, FLXVAL and VARVAL are expressed as a dimensionless number or %. In addition, you can define your own physical units for these variables.

The EQUVAL unit ensures the equality of composition between 2 streams.

4.11 VALVES AND PRESSURE DROP: DPVAL, PIPEVAL

The DPVAL unit allows you to take into account the pressure drop through lines or valves. The pressure drop can be related to the stream flow rate and conditions. Several different equations are available.

The PIPEVAL is a more detailed unit enabling to calculate the pressure drop through wells and lines according to various correlation equations (Colebrook, Von Karman, Blasius, etc.). Heat Transfer to an external media can also be taken into account.

4.12 ON/OFF PROPAGATION

The ON/OFF propagation system of VALI is a quite powerful system enabling to run a model in different operation modes by automatically detecting parts of the process which are not operated at a given time.

The final status of a stream or unit depends on the following succession of mechanisms.

- The status is set according to the status of the PFD, Unit or Stream in the model file.
- This status is then modified according to ON/OFF settings in the measurement files. Combining the syntax UNIT/STREAM/PFD/PRODUCT name ON or OFF, and the @IF/THEN/ELSE keywords you can activate or deactivate parts of the model depending on the measured value of any tag (see [Measurement files](#)).
- The CUTOFF commands is designed to set OFF a material Stream when its measured rate, temperature or pressure is below a given threshold value. The CUTOFF can be set globally for all measurements of a given type (either through the CUTOFF unit or through [Measurement files](#)) or can be set tag by tag via control strings (see [Using control strings](#)) or ValiFilter.
- The STATUS measurement can also set OFF any material Stream via an external binary tag.
- The propagation algorithm will then propagate the OFF status through Units and Streams according to the following rules:
 - if a Unit is OFF, then all its inlet and outlet Streams are set OFF
 - if all the inlet (or outlet) material Streams of a Unit are OFF then the Unit is OFF
 - if only thermal and mechanical Streams are connected to a Unit, it will be set OFF as soon as less than two Streams remains ON
 - On a heat exchanger, if an inlet (or outlet) Stream is OFF then its corresponding outlet (or inlet) Stream is set OFF
 - By exception to the above rules, the TNKVAL unit remains active independently of the status of the connected Streams
 - Flex FILTER functions can be used to set OFF units and streams depending on user-defined rules

Attention:

1. Objects can be set ON until step 2. Starting from step 3, objects can only be set OFF.
2. The CUTOFF always uses the raw measured value, except when the measured value is filtered with the FILTERED syntax in a mea file. In the latter case the FILTERED value of the mea file is used
3. If a tag is OFF or set OFF via a Filter, then the CUTOFF is not executed.

SECTION 5: PROCESS MEASUREMENTS

Measurements are the essential source of information of VALI (in addition to process topology) and cover all information coming either from the plant (temperatures, pressures, rates, on-line analyses, etc.) or from the laboratory (GC, densities, etc.).

The concept of measurement is generalised into pseudo-measurement that represents any numerical information that can be given about the process in addition to actual measurements. For example, turbine efficiencies can be given as pseudo-measurement, although this information comes rather from design data than from a real measurement.

Measurements can be given on streams (e.g.: T, p, rates) or on units (e.g.: pressure drops). They are defined via TAGS, a TAG being a unique name given to a process variable. When needed several tags can be associated to the same variable. Tags are defined either directly in the model or in separate TAG files. Their values can further be changed via MEA files. Finally "control strings" attached to the tag can further modify its value or its accuracy depending on some user defined conditions on the measured value of the tag.

This section covers the following points:

- measurement inaccuracy (or uncertainty)
- measurement redundancy
- handling pressure measurements
- available stream measurements
- available units measurements
- effect of bounds
- validated accuracy
- gross error detection

5.1 MEASUREMENT INACCURACY (OR UNCERTAINTY)

The measurement inaccuracy reflects the confidence given to a measurement value. It indicates the expected range of deviation of the measurement.

Sources of inaccuracies are:

- intrinsic precision of the sensor
- imprecision due to the calibration
- improper location of the sensor
- noise due to small instabilities of operation
- etc.

By assuming a Gaussian distribution of the sensor error, the measurement inaccuracy can be numerically defined on basis of the standard deviation of the distribution. The probability that the real value will be distant from the measured value by less than twice the deviation is about 95 %. Values outside this range will be flagged by VALI as unlikely.

Finding the right combination of the inaccuracies is partly a trial and error procedure. Starting values for the inaccuracies can be the intrinsic imprecision of the sensor as specified by the manufacturer. Some typical values are also proposed in the following sections of this chapter.

In VALI, measurement accuracy is defined on an absolute or a relative basis. In the latter case, the accuracy will be proportional to the measured value and calculated as a percentage of this value, as expressed in the native physical units of VALI. For example a precision of 1% on a temperature of 20C corresponds to 1% of 293.15 K, which is 2.93 K (or C). Take care that relative precision introduced via a TAG file or a MEA file, must be specified as a negative value (-2 meaning 2%).

Some values have a special meaning for VALI:

- 0: the measurement is considered perfect and no correction will be applied to its value
- any value above 10^{10} : the measurement is considered as not available. Setting the accuracy to 1010 has exactly the same effect as removing the measurement from the list of measurements

VALUE	SIGNIFICATION
0	perfect measurement
% (GUI) or <0 (MEA file)	relative deviation (in percentage)
>0	absolute deviation
> 10^{10}	measurement is ignored

TABLE 1 - MEASUREMENT INACCURACIES

Attention: The accuracy of the tags can be made dependant of the measurement range. (See section "[Using Control Strings](#)"). It can also be modified though ValiFilter.

5.2 COVARIANCE BETWEEN MEASUREMENT ERRORS

In some cases, the errors on several different measurements are not totally independent. This can happen for example when the same sensor is used to measure the temperature of several different streams.

In such a case, the measurements errors are not stochastically independent, which is one of the main assumptions underlying the concept of DVR. To overcome this problem, one can introduce an additional element: the **covariance** between measurement errors.

This covariance is introduced as a **correlation coefficient** between the measurement errors, defined as:

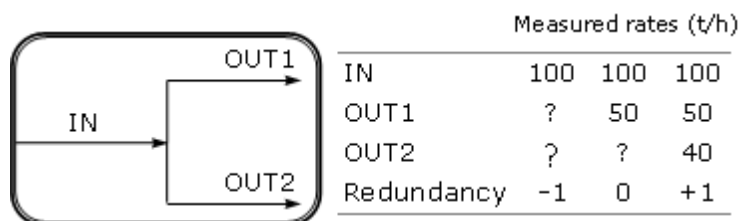
$$\text{correl}_{ij} = \frac{\text{Cov}(x_i, x_j)}{\sigma_i \sigma_j}$$

The value of this correlation coefficient is, on a theoretical basis, bounded between -1 and +1. Positive values indicate that the two measurements will be partly affected by the same errors whereas negative values indicate that part of the error on one measurement affect the other one with the opposite sign.

5.3 MEASUREMENT REDUNDANCY

The redundancy is an essential concept of VALI and is the source of information it uses to correct measurements.

The easiest way to understand the concept of redundancy is to consider an example where only a mass balance is concerned.



If the rate of the stream IN only is measured, then there is not enough information to describe the state of the system, as nothing can determine the quantity going through each outlet lines OUT1 and OUT2. The redundancy is **negative**.

If the inlet rate and one of the outlet rates is known, then we have just enough information to describe the system and the **redundancy level is zero**.

Finally, if both outlet rates are measured in addition to the inlet rate, then we have more information than what is necessary to describe the system. We have a **redundancy of level one**.

The level of redundancy is the number of measurements, which come in addition to the set of measurements needed to be able to calculate the system.

This level is calculated by VALI and when it is negative (sometimes only in a sub-region of the

complete flow sheet), VALI will propose a set of measurements, which could be added.

If it is not possible to add any measurement to the system (because of economical constraints for example), another way of avoiding negative redundancy is to group some units as a more global black box.

5.4 HANDLING PRESSURE MEASUREMENTS

Due to their specific nature, pressure measurements and variables are handled by VALI somewhat differently than other measurements.

The reason for this special treatment comes from the fact that in many cases, pressure measurements cannot be validated because they are not interdependent with other state variables of the streams.

For example, for liquid streams, the pressure has no influence on the stream enthalpy. Therefore, the pressure will not be of any influence on the heat and mass balances that are generated by VALI.

For gas streams (vapour only), the effect of pressure on the enthalpy is non-zero but is, in most cases, very low. Here again, trying to validate pressure measurements on basis of heat and mass balances is difficult if not impossible.

Therefore, it is in most cases only meaningful to validate pressure measurements when pressure drop or pressure equalities equations are available. VALI will therefore request additional measurements when unmeasured pressure variables do not appear in such equations.

Three different ways of handling pressure variables can be specified through the parameter PRESMODE of the OPTVAL unit:

- PRESMODE=0: unit pressure drops (DP) are considered constant when they are not measured
- PRESMODE=1: stream pressure and Unit pressure drops (DP) are considered constant when they are not measured
- PRESMODE=2: stream pressure and Unit pressure drops are considered variable when not measured. (This is the default behaviour)

Pressure measurements can be specified in absolute (bar, atm, Pa, etc) or relative - to atmospheric pressure - basis (barg, MPag, etc.). The reference atmospheric pressure is 1.0 bar by default. This value can be overwritten using ATMOPRES syntax in a measurement file (see "[How to run your data validation model](#)").

5.5 AVAILABLE STREAM MEASUREMENTS

The tables below indicate for each type of stream what variables can be measured. Some typical values are proposed for measurement uncertainty.

THERMAL STREAMS (THRSTRM)

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
LOAD	kW	Heat load	5 - 20 %	
FRACIN	-	Fraction of energy carried out by an inlet stream with respect to all inlet thermal and mechanical streams of a unit		
FRACOUT	-	Fraction of energy carried out by an outlet stream with respect to all outlet thermal and mechanical streams of a unit		

MECHANICAL STREAMS (MECSTRM)

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
POWER	kW	Mechanical power	1 - 5 %	
RPM	RPM	Rotations per minutes ⁽¹⁾		
FRACIN	-	Fraction of energy carried out by an inlet stream with respect to all inlet thermal and mechanical streams of a unit		
FRACOUT	-	Fraction of energy carried out by an outlet stream with respect to all outlet thermal and mechanical streams of a unit		

(1): RPM variable will only be taken into account if it appears in a Flex code, for example by linking the flow rate through the rotating equipment to its rotation speed and pressure drop.

MATERIAL STREAMS (FLOW)

Pressure and temperature related measurements

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
T	C	Temperature	1 - 5 K	relative precision is relative to the absolute temperature (in K)
P	bar	Pressure	1 - 5 %	
H	kJ/kmol	Molar enthalpy	OFF	
HW	kJ/kg	Weight enthalpy	OFF	
SMOL	kJ/kmol.K	Molar entropy	OFF	
VAPF	-	Vapour fraction	1 - 3 %	Valid only for single compound streams
HUMIDITY ⁽¹⁾		Stream humidity (0-100%)	1 - 3 %	H2O must be present in the stream
Z	-	Compressibility factor	OFF	
Gamma	-	Ratio of specific heats (Cp/Cv)	OFF	
A	kJ/kmol	Exergy	OFF	

(1):
$$humidity = PH_{2O} * \frac{P_{stream}}{P_{SAT}}$$

where:

- PH_{2O} is the ratio of the molar flowrate of H_{2O} to the total molar flowrate of the stream;
- P_{Stream} is the pressure of the stream;
- PSAT is the saturated pressure (at the temperature of the stream).

Here is the formula of the saturated pressure:

$$P_{SAT} = \exp(66.9403 - 7469.33/T + 0.474867e-5 \cdot T^2 - 8.03256 \cdot \ln(T))$$

where T is the temperature of the stream (in Kelvin).

Flow rate measurements

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
MOLF ⁽⁴⁾	kmol/s	Total molar flow rate	5 - 10 %	
MASSF ⁽⁴⁾	kg/s	Total mass flow rate	5 - 10 %	
GMOLF	kmol/s	Total dry molar flow rate	5 - 10 %	
GMASSF	kg/s	Total dry mass flow rate	5 - 10 %	
COLDVOLF ⁽¹⁾ (4)	m3/s	Cold liquid volumetric flow (15C)	5 - 10 %	
CONVVOLF ⁽²⁾	m3/s	Cold liquid volumetric flow based on the conventional density (CONVDENS)	5 - 10 %	
VOLF ⁽⁴⁾	m3/s	Volumetric flow rate at flow conditions	5 - 10 %	
HFLOW ⁽³⁾ HFLOWTOT ⁽³⁾ LFLOWTOT ⁽³⁾	kW	Enthalpy flow (by reference to liquid or vapour compound at 25C)	OFF	

(1): $MASSF \text{ (kg/s)} = 1000 \text{ COLDVOLF (m3/s)} \cdot COLDDENS \text{ (kg/l)}$

(2): $MASSF \text{ (kg/s)} = 1000 \text{ CONVVOLF (m3/s)} \cdot CONVDENS \text{ (kg/l)}$

(3): The enthalpy flow (expressed in kW) is defined as the enthalpy carried by a stream by reference to the temperature of 25 C and pressure of 1 atm. For each compound, the reference enthalpy will be either the liquid (or the vapour) if its boiling point is above (or below) 25C. $HFLOWTOT \text{ (kW)} = HFLOW \text{ (kW)} + MOLF \text{ (kmol/s)} \cdot UHVMOL \text{ (kJ/kmol)}$. $LFLOWTOT \text{ (kW)} = HFLOW \text{ (kW)} + MOLF \text{ (kmol/s)} \cdot LHVMOL \text{ (kJ/kmol)}$.

(4): The flow rate can be given for the total flow or for selected phases of the stream (V, L, W, or WL)

Compounds related measurements

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
MRxxx	kmol/s	Partial molar flow rate of compound xxx	OFF	
WRxxx	kg/s	Partial weight flow rate of compound xxx	OFF	
MFxxx	-	Molar fraction of compound xxx	accuracy depends on fraction: ■ > 10%: 1 - 5 ■ 1 - 10%: 0.1 .. 0.5 ■ < 1%: 5 - 20 % ■ ppm(s): 20- 30 %	
WFxxx	-	Weight fraction of compound xxx		
MGxxx	-	Dry molar fraction of compound xxx		
WGxxx	-	Dry weight fraction of compound xxx		
CCxxx	-	Concentration of compound xxx, in g/l (or kg/m3) of liquid at 15°C		
MCxxx	-	Concentration of compound xxx, in g/l (or kg/m3) of liquid at flow conditions		
LFxxx	-	Liquid volume fraction of compound xxx		
LGxxx	-	Dry liquid volume fraction of compound xxx		

Attention:

1. If a compound fraction is measured as 0, then its accuracy is automatically set to zero (perfect measurement).

2. Specifying a fraction of 100% as a measurement is only allowed if the measurement is set constant and all the other compounds are either not measured or set to zero.
3. If a compound only appears as traces (ppm), then setting it to zero is usually preferred (as it will not affect the balance). The compound should only be considered if it is necessary for an adequate modelling of the downstream part of the process.

Density measurements

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
DENSITY	kg/m ³	Density at flow conditions	1 %	
COLDDENS	D15/4	Liquid Gravity at ambient conditions (can be 15C/4C, 25C/4C or 60F/60F)	1 %	
API	API	API Gravity	1 %	
CONVDENS	D15/4	Conventional Liquid Gravity (similar to COLDDENS but independent of composition)	1 %	
MOLW	kg/kmol	Molecular weight	1 %	

Distillation curve measurements

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
TBPVxxx	C	TBP curve: temperature at xxx vol%	3 K	
TBPWxxx	C	TBP curve: temperature at xxx wt%	3 K	
D86Vxxx	C	D-86 curve: temperature at xxx vol%	3 K	

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
D86W _{xxx}	C	D-86 curve: temperature at xxx wt%	3 K	
D60V _{xxx}	C	D-1160 curve: temperature at xxx vol%	3 K	
D60W _{xxx}	C	D-1160 curve: temperature at xxx wt%	3 K	

Heating value measurements

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
LHVWT	kJ/kg	Lower heating value on a weight basis	1 %	
LHVMOL	kJ/kmol	Lower heating value on a molar basis	1 %	
UHVWT	kJ/kg	Upper heating value on a weight basis	1 %	
UHVMOL	kJ/kmol	Upper heating value on a molar basis	1 %	

Specific measurements for oil and gas production

In the below table, standard conditions are defined as 15 C and 1 bar.

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
QOIL	m ³ /s	Oil flow rate at standard conditions	OFF	
QGAS	m ³ /s	Gas flow rate at standard conditions	OFF	
QWATER	m ³ /s	Water flow rate at	OFF	

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
		standard conditions		
QLIQ	m3/s	Liquid flow rate at standard conditions	OFF	
GOR	-	Gas to oil ratio at standard conditions	OFF	
CGR	-	Condensate to gas ratio at standard conditions	OFF	
WCT	-	Water to liquid ratio at standard conditions (water cut)	OFF	
GLR	-	Gas to liquid ratio at standard conditions	OFF	
GVF	-	Gas volume fraction at flow conditions	OFF	
WLR	-	Water to liquid ratio at flow conditions	OFF	
DENSG	kg/m3	Density of the gas phase at flow conditions	5 kg/m3	
DENSO	kg/m3	Density of the oil phase at flow conditions	5 kg/m3	
DENSW	kg/m3	Density of the water phase at flow conditions	5 kg/m3	

Miscellaneous tags

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
STATUS		Stream ON/OFF status 1,OPEN,OPN,OUVERT, Ouvert => ON		Enables to set a stream OFF via a binary tag

NAME	PHYS. UNITS	DESCRIPTION	TYPICAL ACCURACY	REMARKS
		0,CLOSED,CLSD,FERME, Ferme => OFF		
PRODID		Product ID that defines the product associated to the stream		The list of products IDs and codes must be made available in a DICTIONARY section through a mea file

Attention: User-defined measurements can also be used on streams by registering them as a "model measurement" and defining the way to calculate them through a Flex code. (See "[Model measurements page](#)" and "[User-defined stream measurements](#)")

5.6 AVAILABLE UNIT MEASUREMENTS

Many UNIT parameters can be introduced as measurement. Table below lists some of the most used unit measurements. Refer to the unit's description for more information.

NAME	DESCRIPTION	TYPICAL ACCURACY	UNIT TYPES
DPi	Pressure drop	10 - 30 %	All units featuring pressure drop equations
CFDPi	Pressure drop correction factor	5-10 %	
EXTENTi	Reaction extension		RCTVAL, RLVVAL, RKPVAL
DTEQi	Delta temperature from equilibrium		BWHVAL, RKPVAL
EFFIC	Isentropic efficiency Volumetric Efficiency	2 - 10 %	COMVAL, TURVAL PUMVAL
POLYEFFI	Polytropic efficiency	2 - 10%	COMVAL, TURVAL
A	Heat exchange area	0	HEXVAL, PRHVAL, ECOVAL
U	Heat exchange coefficient (convection)	5 - 20 %	HEXVAL, PRHVAL, ECOVAL HEXVAL
G	Heat exchange coefficient	5 - 20 %	

	(radiation)		
LOAD	Heat load	2 - 10 %	HEXVAL, PRHVAL, ECOVAL

5.7 EFFECT OF BOUNDS

THE SOLVER BOUNDS

Each variable used in a VALI model has at least one lower and one upper bound, called the solver bounds.

These solver bounds have a default value which depends on the type of value (temperature, pressure, etc) and the optimiser used to solve the DVR problem (SOLDOG or SQP-IP). For SOLDOG, which has not been designed for handling bounds, the bounds are quite large while for SQP-IP, somewhat restrictive values are used. In addition, a special mode (activated through the OPTVAL unit) allows SOLDOG to accept negative values for partial flow rates when the corresponding streams are not involved in thermodynamic calculations.

These default values can be overwritten by using a special mathematical unit of type [BNDVAL](#) where the user can enter its own lower and higher bounds for temperatures, pressures, etc. These user bounds can be more or less restrictive than the default bounds.

BOUNDS DEFINED ON TAGS

When defining the measurements through their tags, bounds are also defined on the corresponding variables.

Three levels are considered for both low and high bounds:

- Warning (L -> H): a warning message will be issued if the measured or the validated values are outside this range
- Error (LL -> HH): an error message will be issued if the measured or the validated values are outside this range. These bounds are also used by ValiFilter
- Solver (SL -> SH): a fatal error occurs if the measured value is outside this range. The validated value will be kept between these bounds. Solver bounds must always be within the default bounds (possibly modified by a BNDVAL unit)

At convergence, a detail analysis is performed by VALI to determine which variables are on their bounds. Each variable can be in three states:

- one of its **solver bounds** (either High or Low) **is active**. This type of bound is reported as Solver bounds in the report. Changing the value of the bound will change the results of the model. In some cases, several bounds are related and redundant, for example, several related flow rates might go to zero simultaneously. The redundant bounds are flagged.
- its value is **close to one of the bounds**, i.e. the distance from the bound is lower than its validated accuracy multiplied by a factor (which is 1 by default and can be overwritten through the OPTVAL unit). It means that there is a significant probability that the actual value would be on its bound. If the validated accuracy is large, it might indicate a lack of information for determining the actual value of the variable.
- its value **is not influenced by the bounds**.

BOUNDS CONTROL

The activation of solver bounds may in some cases lead to model instability and/or to numerical singularities due to degenerated equations. The mechanism of bound control keeps the variable at an adequate distance from the bounds avoiding any instability or singularity. This mechanism can be used with any of the two solvers.

Attention: When using SOLDG solver together with bounds control mechanism it is recommend to set the parameter CONSTR of unit OPTVAL to 1.000001, to avoid staying to much away of the controlled bound, which would inhibit the bound control mechanism.

5.8 VALIDATED ACCURACY AND COVARIANCE

At convergence of the validation problem and after linearization, the validated values of the measured and unmeasured variables are expressed as a linear combination of the measured values. Using error propagation computation, it is then possible to evaluate the validated accuracy of the validated values on basis of these coefficients and of the accuracy of the measurements.

Variables (either measured or unmeasured) which are activating their bounds are considered as constants when calculating the validated accuracy of the other variables.

In practice, the validated accuracy (in absolute value) is always lower or equal to the accuracy of the related measurement. If the two values are equal, than it indicates a variable which in fact cannot be validated.

The covariance between the validated values is calculated through a separate tool, the sensitivity analysis. This covariance expressed as a correlation factor indicates measurements for which the errors are related.

5.9 GROSS ERROR DETECTION AND ELIMINATION

On basis of the redundancy level of the DVR problem, statistical tests enable the detection of possible gross errors using the Chi-square test value and optionally an additional individual test.

The **Chi-square test** (or Chi2) value depends on the number of redundancies of the system, active bounds being considered as adding new levels of redundancy, and on the statistical certainty of the test, typically 95% (see equation 126 of VDI-2048). If the weighted sum of penalties is higher than the Chi-square test value, then there is a significant suspicion that gross errors exist.

Attention: The Quality of a run is defined as the ratio between the weighted sum of penalties and the Chi2 value. The Quality should thus be less or equal to 1.

The **individual test** is based on the penalty associated to each of the measurement. If this penalty is above a given threshold, which again depends on the statistical certainty of the test, then again the measurement may be in gross error. The individual test can be based on:

- the penalty of the measurement, as calculated in the objective function: measurement correction weighted by the measurement uncertainty
- the penalty of the correction, as recommended by the VDI-2048: measurement correction weighted by the correction uncertainty
- the impact of the measurement, which is the decrease of the total penalty when removing the corresponding measurement

The latest test is the most accurate in detecting the existence and the location of gross errors.

If any of the two tests is positive, all results obtained with the model are to be used with care: validated values, identified performance factors and their validated accuracy.

The **automatic procedure of elimination** of gross errors of VALI can be activated by adding the mathematical unit GROSSERR in the main PFD. This unit allows controlling the procedure by which suspect measurements are progressively eliminated from the set of data until the statistical tests indicate that no gross errors are present.

The selection of the tags to be eliminated can be based on any of the three individual tests listed here above.

The **Highest Impact** procedure is the most powerful method. It detects the impact on the total sum of penalties of removing each of the measurements. This impact is then weighted (by simple multiplication) by the Failure probability factor associated to each measurement. The measurement with the highest weighted impact is then removed and the procedure is iterated until the Chi2 test is satisfied or, depending on the user choice, until the individual test is satisfied as well.

Eliminated tags are flagged by a distinctive colour in ValiModeller GUI PFDs as well as in the reports. A pseudo-penalty is calculated based on the precision of the measurement (or of the correction in

VDI compliant mode).

The parameters of unit GROSSERR enable to select various detection/elimination strategies, but as said above, the Highest Impact procedure is recommended in all cases.

More information is given on the Gross Error Elimination procedure in the description of unit [GROSSERR](#).

SECTION 6: PETROLEUM REFINING APPLICATIONS

The data validation of processes involving petroleum cuts involves two steps:

- step 1: identification of pseudo-compounds, using only mass balances
- step 2: complete heat and mass balance of the process

6.1 THE CONCEPT OF PSEUDO-COMPOUND

Crudes and petroleum cuts are mixtures of an incalculable number of chemical compounds. Their composition can therefore not be based on fractions of well-identified chemical compounds.

Specific analyses are available to describe these cuts, among which the distillation curves and the density are the most important. In addition, a breakdown of some chemical compounds is sometimes available for the light-ends (typically C1 to C5). Today, such a breakdown also covers heavier compounds (up to C8 or C10) more and more, but compounds of the same family and number of carbons are then lumped together.

On the modelling side, it is common practice to model such streams by representing them as a mixture of a limited numbers of so-called petroleum fractions or pseudo-compounds.

These pseudo-compounds are defined by their boiling point and by their density. According to the crude type and origin, different pseudo-compounds must be used to get an accurate representation. The usual way of doing this is to select a number of compounds with given boiling point and to identify their fraction and their density in order to match the stream distillation curve and density.

When the boiling point and the density of the compounds are known, their other thermodynamic properties (like their critical properties or specific heat) can be calculated using standard correlations as proposed for example by American Petroleum Institute (API).

On basis of this identification, it is then possible to perform various heat and mass balances and other thermodynamic calculations.

Pseudo-compounds are available in the library CRUDE by step of 5 °C. A typical Thermod named CRUDE is defined in the Refining.bls file with a typical mix of pseudo-compounds.

6.2 DATA VALIDATION OF DISTILLATION CURVES

When several petroleum cuts are involved in a distillation process, the inlet and outlet streams of the column must obviously use the same pseudo-compounds. This problem can be expressed as a DVR system involving:

- as variables, the pseudo-compounds densities and their fractions in each stream
- as equations, the mass balances of the distillation column for each pseudo-compound
- as measurements, the densities and distillation curves of the streams

Having solved this system of equations, one can then estimate the other thermodynamic properties

of the pseudo-compounds. Having identified these properties it is then possible to use these compounds in more general DVR problems, involving heat balances and liquid/vapour equilibrium laws.

6.3 USING THE CRUVAL UNIT

The CRUVAL unit has been developed to solve this kind of problem. Several CRUVAL units can appear in a model and they can be combined with other DVR units (BBXVAL, SPLVAL, etc.).

When the pseudo-compounds identification is activated (IDENT=1), only mass balances are performed and some of the pseudo-compounds properties become variables of the system and are therefore identified. At the end of the run the other pseudo-compounds properties are identified and saved in the model file.

Attention: When in identification mode, you should only work with mass balances. The use of reactors is also prohibited.

When the pseudo-compounds identification is deactivated (IDENT=0), these pseudo-compounds properties are considered as constant. There is then no limitation in the use of the CRUVAL unit in combination with any other VALI unit.

6.4 USING DISTILLATION CURVES MEASUREMENTS

Information on the densities and distillation curves of the streams is given as usual measurements. These measurements can be combined with composition measurements, for example to specify light ends.

Three types of distillation curves are available: TBP, ASTM D-86 and ASTM D-1160. Each of them can be expressed on a weight basis (TBPW, D86W and D60W) or on a volume basis (TBPV, D86V and D60V). Each point is given as a separate measurement:

Example:

- TBPW005: defines the measurement of the 5 wt% point temperature
- D86V050: defines the measurement of the 50 vol% point temperature

When at least three points of a distillation curve are given for a stream, VALI will do the following:

- if not given the initial (0 %) and final (100 %) TBP points are extrapolated
- the partial rates of compounds which fall out of this boiling range are set equal to zero, except if a composition measurement is already defined
- some additional constraints are added so as to keep the TBP curve as smooth as possible
- the TBP and D86 curves will be added to the usual Stream report in the <model>.r4v file

The measurements of densities and distillation curves should be kept in both steps

More information is given on this topic in the description of unit [CRUVAL](#).

Attention: Distillation curves of different types (e.g. TBP and D86) cannot be defined as measurements on the same stream. By defining a dummy black box with one inlet and one outlet stream, you can however define one type of distillation curve on the inlet and a second type on the outlet.

SECTION 7: COMPATIBILITY WITH VDI-2048

7.1 THE VDI-2048 GUIDELINE

The VDI 2048 guideline (October 2000 edition, available from Beuth Verlag GmbH, 10772 Berlin) defines the fundamentals of "Uncertainties of measurement during acceptance tests on energy-conversion and power plants". Its objective is "(...) to give instructions (...) for the evaluation of acceptance measurements in terms of an informative comparison of guaranteed characteristics in consideration of the special conditions in energy-conversion and power plants".

7.2 THE VDI MODE OF VALI

Although the mathematical principles of VALI are in total conformity with the recommendations of the VDI-2048 guideline, a number of differences do exist between the formulation of VDI-2048 and the default operating mode of VALI.

These differences concern:

- the definition of the accuracy, based on the standard deviation s in VALI and on the 95% confidence level in VDI (1.96σ)
- the definition of the penalty of individual measurements, which is based on the measured accuracy for VALI and on the correction accuracy for VDI
- the way reports are generated. The common reports on units and streams are replaced by the mass and energy balance of the units and a more specific report for the streams (file r4v)

To overcome these differences, a number of settings have been added to the list of options (units OPTVAL and REPVAL) so that VALI strictly conforms to the guideline.

A global runtime option (-VDI) automatically activates all these settings in one shot.

Chapter 6 (assessment of the fulfilment of a guarantee) of the VDI 2048 can be fulfilled by introducing additional equations via the FLEX language (see [Flex Language description](#) chapter). Values calculated via Flex language will be presented in the standard reports of VALI (r4v, r6v, etc.).

7.3 COMPLIANCE TO THE VDI-2048 GUIDELINE

VALI, when operated in its VDI mode, is fully compliant with the VDI-2048. This compliance has been officially stated by VDI representatives.



belsim

CHAPTER 3

THERMODYNAMIC CALCULATION METHODS

SECTION 1: INTRODUCTION

The methods depend on the chemical compounds involved, also on process temperature and pressure ranges. The following values can be calculated for liquid and vapour phases : fugacities, enthalpies and entropies, volumes, thermal conductivities, viscosities.

Solid chemical compounds may be considered, though they cannot be involved in phase changes.

Section 2 describes the calculation methods.

When considering mass balances only, the default ideal methods proposed by VALI are recommended.

When the data validation also involves phase equilibrium, appropriate methods must be selected for fugacity calculations. This is because VALI determines phase equilibrium by equalising the fugacities of each chemical compound in both phases.

When considering energy balances, appropriate methods must be selected, for enthalpy calculations and for phase equilibrium calculations if phase changes occur.

Section 3 describes the fundamental properties of chemical components available in the data-bases delivered with VALI

FLUIDS for liquid-vapour chemical compounds

SOLIDS for solid chemical compounds

PETROL for pseudo-compounds used to model petroleum fractions

FUELS for pseudo-compounds used to model fuel-oil and coal

Section 4 describes the interaction parameters used in the calculation methods available in the BELSIM environment.

Section 5 shows how to choose the appropriate methods for some typical mixtures and gives advice on this subject.

Section 6 lists the references.

Sections 7 and 8 give the list of fluids and solids compounds.

Section 9 indicates how to handle user-defined compounds.

Section 10 describes how to use third parties thermodynamic packages compliant with the GCO standard.

Section 11 discusses about the uncertainty of thermodynamic methods.

SECTION 2: BELSIM CALCULATION METHODS

Methods are available to calculate:

- the fugacity of a chemical compound in the liquid and vapour phase (This property is used to calculate phase equilibrium)
- the enthalpy and entropy of liquid and vapour mixtures
- the molar volume of liquid and vapour mixtures
- the thermal conductivity of liquid or vapour mixtures
- the viscosity of liquid or vapour mixtures

Attention: Methods are not available to calculate the thermal conductivity or the viscosity of bi-phasic mixtures (liquid plus vapour).

Thermodynamic properties calculation methods are selected from the method page of the Thermod Notebook:

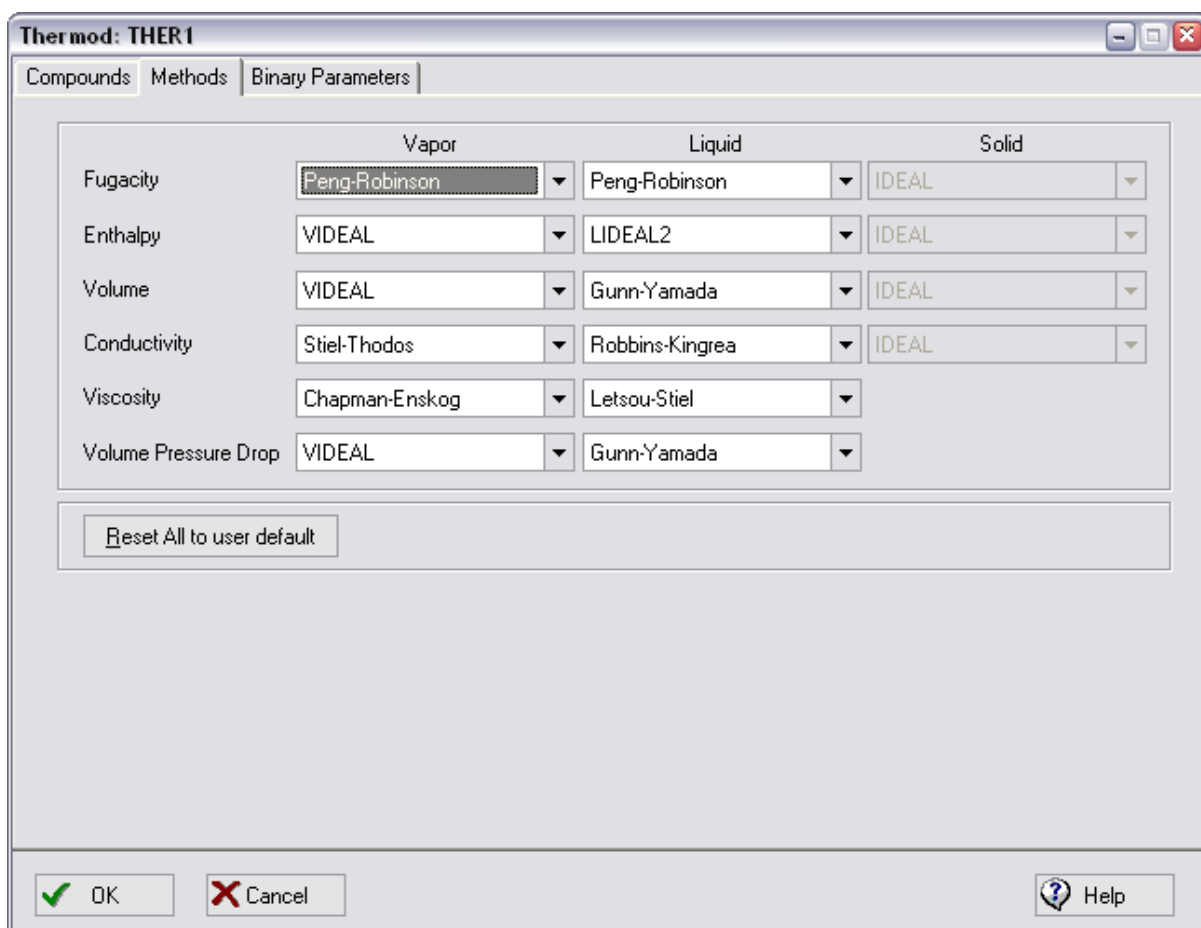


FIGURE 1 - THE METHOD PAGE OF A THERMOD NOTEBOOK

2.1 FUGACITY CALCULATION METHODS FOR THE VAPOUR PHASE

For the **vapour phase**, the simplest calculation method corresponds to a mixture of perfect gases whose compound fugacities are equal to their partial pressures: **VIDEAL** (default method)

For the **vapour phase**, methods based on the equation of state are available:

- LASSC
- LEE-KESLER
- PENG-ROBINSON
- REDLICH-KWONG
- SOAVE
- VIRIAL

These methods consider the effects of binary interactions between identical and different molecules.

Except for the **REDLICH-KWONG** method, the interaction parameter values must be provided, or the default values provided will be used. Refer to section "[Interaction parameters](#)" for more information.

For **pure water** three special methods are available :

- IAPWS-IF97: steam tables according to norm 97 of IAPWS
- VDI: based on VDI tables (Verein Deutscher Ingenieure)
- NBS: equation of state of the National Bureau of Standards

2.2 FUGACITY CALCULATION METHODS FOR THE LIQUID PHASE

For the **liquid phase**, the simplest calculation method is the **LIDEAL** (default method)

It corresponds to an ideal liquid mixture whose compound fugacities are calculated by the Raoult law. The fugacity of supercritical compounds is calculated by the following **LYCKMAN** empirical function :

$$\ln\left(\frac{f}{P_c}\right) = 5.42 * \left(1 - \frac{T}{T_c}\right)$$

Where:

- T_c and P_c are respectively the critical temperature (K) and pressure (bar) of the pure component
- T is the temperature (K)
- f is the fugacity (bar)

Some methods are based on the liquid phase equation of state:

- LASSC
- LEE-KESLER
- PENG-ROBINSON
- SOAVE
- REDLICH-KWONG

These methods consider the effects of binary interactions between identical and different molecules. The interaction parameters values must be provided or the default values will be used. Refer to section "[Interaction parameters](#)" for more information.

Other methods are based on the principle of corresponding-states :

- BRAUN K10
- CHAO-SEADER
- GRAYSON-STREED

These methods do not require interaction parameters values and are mostly used for hydrocarbon mixtures.

Another group of methods is based on free energy expressions for the liquid mixture :

- NRTL
- UNIQUAC

For these, it is a must to provide interaction parameters values. Refer to section "[Interaction parameters](#)" for more details. The activity coefficients of each compound are calculated from these free energy expressions. The fugacity is then obtained by multiplying them by the Poynting correction for pressure effect, by the pure compound reference fugacity and by the molar fraction.

The solubility of gases in a liquid mixture may also be considered but information must be provided to calculate infinite dilution activity coefficients.

- UNIFAC

This method is similar to UNIQUAC but there is no need to provide numerical interaction parameters. The interaction parameters are defined by the model from structural information about pure components (group contributions) : it is just necessary to define the list of typical groups

associated with each component.

For **pure water** use :

IAPWS-IF97 or NBS or VDI

2.3 CALCULATION METHODS FOR ENTHALPY AND ENTROPY

For the **vapour** phase, the simplest calculation method is the VIDEAL (default method)

This method is based on the ideal gas heat capacity. When water is present in the vapour mixture, this method considers the real properties of water to calculate the mixture enthalpy and entropy.

The PTC4 and PTC42007 methods are dedicated to fumes.

For the **liquid phase**, two simple calculation methods are:

- LIDEAL (default method)
- LIDEAL2
- QUADRAT1
- QUADRAT2

The LIDEAL method calculates the liquid enthalpy by subtracting the vaporisation enthalpy calculated by the Watson method from the vapour enthalpy of the mixture calculated with the VIDEAL method.

The LIDEAL2 method calculates the liquid enthalpy by subtracting the vaporisation enthalpy, at the normal boiling point, from the vapour enthalpy of the mixture calculated with the VIDEAL method, then calculating the temperature effect with the liquid heat capacity expression.

When water is present in the liquid mixture, the above mentioned methods consider the real properties of water to calculate the mixture enthalpy and entropy.

QUADRAT1 and QUADRAT2 calculates a deviation from respectively LIDEAL and LIDEAL2 methods by adding a quadratic mixing enthalpy calculated as:

$$DH_{mix} = \sum ijENTH \cdot x_i \cdot x_j$$

Where:

- $ijENTH$ is the binary parameter, expressed in kJ/kmol
- x_i and x_j are the molar fractions of the chemical compounds i and j in the liquid phase

For pure water the IAPWS-IF97, NBS and VDI methods are available.

For liquid and vapour phases, a group of methods, based on the equation of state, is applicable:

- LASSC
- LEE-KESLER
- PENG-ROBINSON
- SOAVE
- REDLICH-KWON

The interaction parameters for enthalpy and entropy calculations are the default values proposed, unless customised values were entered previously and the same method as for fugacity calculations is used. In this case VALI will use the interaction parameters entered for fugacity calculations.

For nonpolar **hydrocarbon** mixture and for both phases, the method CURL-PITZER may be used . It is based on the principle of corresponding-states.

Attention: For ALL methods of ENTHALPY calculations (except VDI and IAPWS-IF97, for which liquid state at 0 C is the reference), the reference state is the ideal gas at 298.15 K. Because of the difference of reference state, streams based on VDI or IAPWS-IF97 cannot be mixed with other streams in most units. The only exception is the BBXVAL unit where a mixing can be carried out with the following limitation: all inlet OR outlet streams must have the same reference state.

2.4 CALCULATION METHODS FOR MOLAR VOLUME

For liquid or vapour phases, the simplest methods are respectively :

- GUNN-YAMADA:Liquid phase (default method)
- IDEAL: Vapour phase (default method)

They are based respectively on the Gunn and Yamada correlation and on the perfect gas law.

Methods based on the Wuithier correlation are also available for hydrocarbon mixtures . In these methods the mixture density is calculated on basis of its "cold density" (density at 15°C) and its actual temperature.

- Wuithier colddens: the density of the mixture at 15°C is calculated on basis of the composition and the compounds density (COLDDENS property)
- Wuithier convdens: the density of the mixture at 15°C is calculated on basis of a conventional density which must be measured (CONVDENS)

Methods based on equations of state are also available .

For both **liquid and vapour** phases :

- LASSC
- LEE-KESLER
- PENG-ROBINSON
- SOAVE
- REDLICH-KWONG

Attention: The LASSC equation of state method is recommended, since it is more accurate than the other methods.

For the **vapour** phase: VIRIAL.

The interaction parameters for molar volume calculations are the default values, unless customized values were entered previously and the same method as for fugacity calculations is used. In this case VALI will use the interaction parameters entered for fugacity calculations.

For nonpolar hydrocarbon mixtures, you can use, for both phases, the CURL-PITZER method, which is based on the principle of corresponding-states.

For **pure water** the IAPWS-IF97, NBS and VDI methods are available.

2.5 CALCULATION METHODS FOR THERMAL CONDUCTIVITY

For the **vapour** phase, two methods may be used :

- PEDERSEN
- STIEL-THODOS (default method)

The first applies the principle of corresponding-states and is valid for nonpolar hydrocarbon mixtures.

The second one is based on the Stiel and Thodos correlation associated to the Wassiljew a mixing rule.

For the **liquid phase**, three methods may be used :

- PEDERSEN
- POLYNOM (default method)
- ROBBINS-KINGREA

The first, based on the principle of corresponding-states, is valid for nonpolar hydrocarbon mixtures.

The second is based on empirical regression coefficients.

The third one, based on the Robbins and Kingrea correlation, is used for any reduced temperatures lower than 1.0. It is associated to the mixing rule of Li.

For **pure water** the IAPWS-IF97, NBS and VDI methods are available.

2.6 CALCULATION METHODS FOR VISCOSITY

For the **vapour** phase, you have a choice of two methods :

- CHAPMAN-ENSKOG (default method)
- PEDERSEN

The first method applies the correlation of Chapman and Enskog associated to the mixing rule of Brokaw. You can use it for polar and nonpolar chemical compounds mixtures.

The second one is based on the principle of corresponding-states and is valid for nonpolar hydrocarbon mixtures.

For the **liquid** phase, three methods may be used :

- LETSOU-STIEL
- PEDERSEN (default method)
- VAN-VELZEN

The first, based on the Letsou and Stiel correlation, is used for all reduced temperatures to avoid discontinuities at 0.75, even though this correlation is really only applicable for reduced temperatures between 0.75 and 1.

The second method, based on the principle of corresponding-states, is applicable for nonpolar hydrocarbon mixtures.

The third applies the correlation of Van Velzen, Cardozo and Langenkamp, which is normally valid for reduced temperatures lower than 0.75. To avoid discontinuities in the liquid viscosity calculation, the correlation is extrapolated up to the critical temperature.

The logarithmic mixing rule is used for the first and the third method.

For **pure water** the IAPWS-IF97, NBS and VDI methods are available.

2.7 CALCULATION METHODS FOR SOLID COMPOUNDS

Solids may be present in a mixture but they cannot participate in phase changes. Their enthalpies will automatically be calculated by heat capacity coefficients. The viscosity calculation of mixtures containing solids does not take their presence into account. The solid contribution to the molar volume and the thermal conductivity is simply added to the mixture property.

2.8 PHASE EQUILIBRIUM CALCULATIONS

Phase equilibrium calculations are realised by equalising the fugacities of each chemical compound in each phase. **The number and types of phases** that VALI has to take into account has to be specified.

This is done by the means of the Stream State parameter which can take one of the following values :

- L: when the mixture is liquid. There will be no phase equilibrium calculation,
- V: when the mixture is vapour. There will be no phase equilibrium calculation
- S: when the mixture is solid. There will be no phase equilibrium calculation
- LV: when the mixture can be liquid and/or vapour. Liquid-vapour phase equilibrium calculations are systematically performed to check the thermodynamic state of the mixture
- LLV: when two liquid phases and a vapour phase can coexist. Liquid-liquid-vapour phase equilibrium calculations are systematically performed to check the thermodynamic state of the mixture
- LL: when two liquid phases can co-exist. Liquid-liquid phase equilibrium calculations are systematically performed to check the thermodynamic state of the mixture
- WL: when one hydrocarbon liquid phase (without water), one aqueous liquid phase and a vapour phase can co-exist, Liquid-liquid-vapour phase equilibrium are systematically calculated to check the thermodynamic state of the mixture
- WL: when one hydrocarbon liquid phase (without water) and one aqueous liquid phase can co-exist. Liquid-liquid phase equilibrium are systematically calculated to check the thermodynamic state of the mixture

Using a multiphase "Stream State" parameter means that VALI must check their existence but does not mean that these phases exist.

For instance, with the "Stream State" set to "LV", VALI will conclude that the mixture :

- is in the liquid state if its temperature is below the bubble point
- is in liquid-vapour state if its temperature lies between the bubble and the dew point
- is in the vapour phase if its temperature is higher than its dew point

Attention: With the "Stream State" set to "LV", VALI will never check for the existence of two liquid phases.

It is essential that you specify properly this "State" parameter during Stream definition.

The enthalpy of a multiphase mixture is the weighted sum of the phase enthalpies, the weight factors being obtained from the phase equilibrium calculation. The same rule applies for the entropy and the volume.

When a solid is present in the mixture, it is considered in the liquid phase and contributes to the enthalpy, entropy and volume of the liquid phase.

SECTION 3: FUNDAMENTAL PROPERTIES OF CHEMICAL COMPOUNDS

Several BELSIM standard data bases are delivered with VALI: FLUIDS, SOLIDS, PETROL and FUEL.

During the definition of a Compound, ValiModeller copies the fundamental properties from the BELSIM standard data base into the Process Data Base. When some data are missing, default values are generated.

3.1 AVAILABLE DATA BASES

THE FLUIDS DATABASE

This BELSIM database contains the fundamental properties needed to describe the behaviour of about 830 chemical compounds in the liquid and/or vapour state.

The available data includes:

- the critical coordinates
- the heat capacity coefficients of the ideal gas
- the saturation curve coefficients
- etc.

THE SOLIDS DATABASE

The BELSIM standard data base contains the fundamental properties necessary to describe the behaviour of about 650 chemical compounds in the solid state.

The available data include :

- the heat capacity coefficients for the solid phase
- the coefficient of the sublimation curve
- etc.

THE PETROL DATABASE

This data base contains the definitions of Petroleum fractions with boiling points increasing by steps of 5 C.

These pseudo-compounds can be used in combination with other compounds. Their properties can be identified to match cut densities, by using a unit of type CRUVAL.

THE FUELS DATABASE

This data base contains the properties of pseudo-compounds used to model fuel-oil and coal. For fuel-oil, 4 compounds can be mixed to adjust the Sulphur, the Hydrogen and the Water contents,

whereas for coal, you can in addition adjust Oxygen, Nitrogen and Ashes contents:

- Fuel-oil: C-FUEL, H-FUEL, S-FUEL, H2O-FUEL
- Coal: C-COAL, H-COAL, S-COAL, H2O-COAL, N_COAL, O_COAL, ASH_COAL

Typical compositions are presented in the table below.

	C	H	S	H2O	N	O	ASH	LHV	DENSIT
	WT%	WT%	WT%	WT%	WT%	WT%	WT%	KJ/KG	Y
									KG/M3
Light Fuel Oil	86.65	13.13	0.23	0.00				42803	852
Heavy Fuel Oil	86.17	11.32	1.14	1.37				40634	945
Coal 1	86.07	4.93	1.0	8.0	0	0	0	34199	
Coal 2	89.14	4.86	1.0	5.0	0	0	0	35121	
Coke	97.61	0.79	1.0	0.6	0	0	0	32999	

TABLE 1 - TYPICAL COMPOSITIONS OF FUELS

Compound properties can be reviewed in the Data page of the Compounds Notebook.

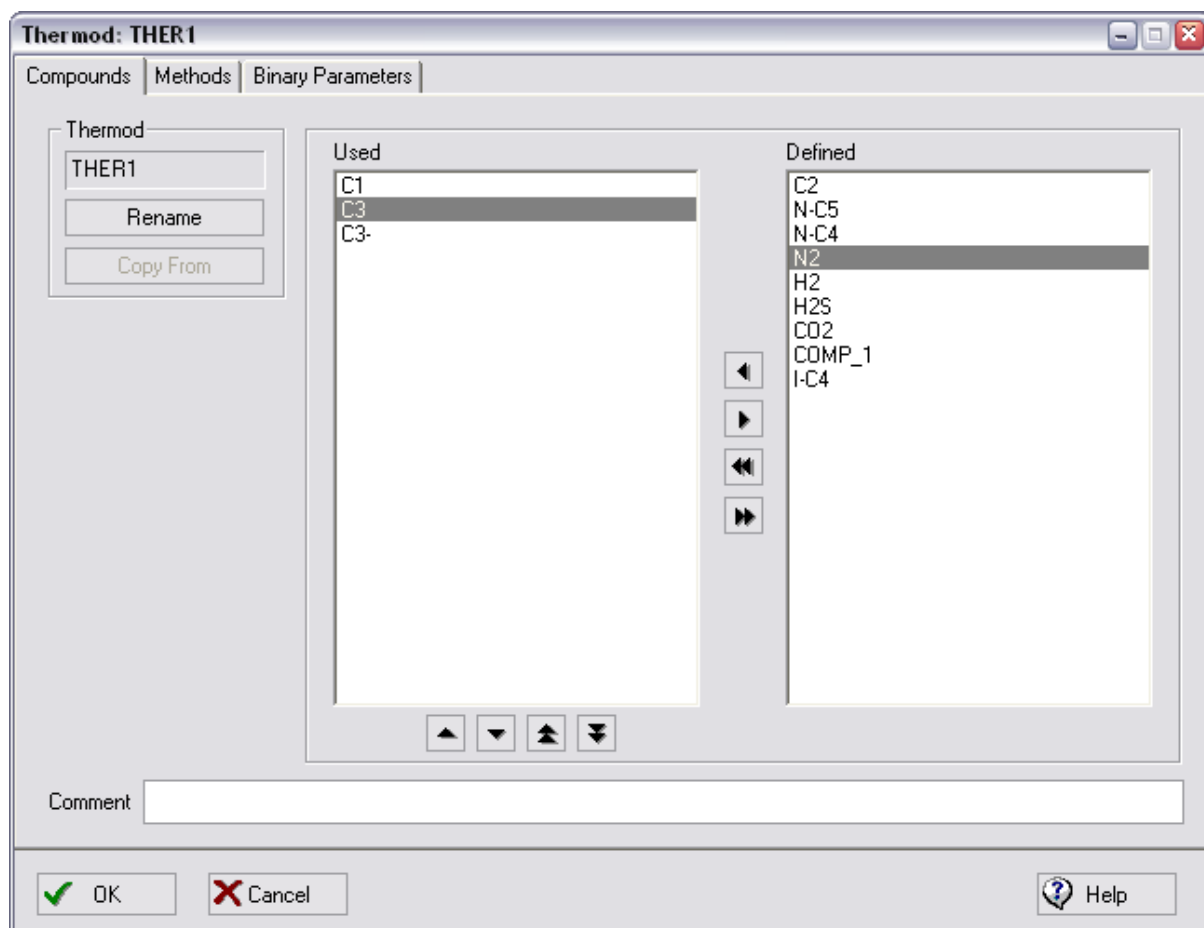


FIGURE 2 - THE DATA PAGE OF THE COMPOUNDS NOTEBOOK

3.2 DATA NEEDED FOR LIQUIDS AND VAPOURS

FUGACITY CALCULATIONS

Table below presents, for each listed calculation method, the set of fundamental properties, whose values are required for each chemical compound of the mixture.

The critical coordinates are always needed to estimate starting values for the phase equilibrium calculations.

						P		P		G	A	
						A		A		Y	S	
						R		R	D	R	S	
						O	L		S	I	A	O
						M	A	P	V	O	P	D
						E	S	V	S	L	O	I
	T	T	P	V	G	S	A	A	U	L	U	A
	B	C	C	C	A	C	P	T	B	M	S	R
VIDEAL												
LIDEAL		X	X				X					
VIRIAL		X	X						X	X	X	
PENG-ROBINSON		X	X		X							
LASSC		X	X	X	X	X						
SOAVE		X	X		X							
REDLICH-KWONG		X	X									
LEE-KESLER		X	X	X	X							
CHAO-SEEDER		X	X		X			X	X			
GRAYSON-STREED		X	X		X			X	X			
BRAUN K10	X	X					X					
NRTL		X	X		X		X	X				
UNIQUAC		X	X		X		X	X				X
UNIFAC		X	X		X		X	X				X

PROPERTY	DESCRIPTION
TB	Normal boiling temperature (K)
TC	Critical temperature (K)
PC	Critical pressure (bar)
VC	Critical volume (m3)
PARLASSC	Parameter for LASSC equation of state
OMEGA	Acentric factor

PROPERTY	DESCRIPTION
PVAP	Boiling curve coefficients
VSAT	Molar volume at reference temperature TREF
PARSOLUB	Solubility parameter of Hildebrand
DIPOLM	Dipole moment
GYRADIUS	Gyration radius
ASSOCPAR	Associative parameter
RQVDW	Van der waals parameters

TABLE 2 - DATA NEEDED FOR FUGACITY CALCULATIONS OF LIQUIDS AND VAPOURS

For the UNIFAC method, it is also necessary to introduce structural information about components: interaction parameters are defined from structure of each component of the mixture.

ENTHALPY AND ENTROPY CALCULATIONS

Table below presents, for each listed calculation methods, the set of fundamental properties, whose values must be present in the model file for each chemical compound of the mixture.

Formation data is used to calculate reaction heats or reaction free energy variations when chemical reactions are to be taken into account.

PROPERTY	DESCRIPTION
H0F G0F	Enthalpy and free energy of formation
TB	Normal boiling temperature (K)
TC	Critical temperature (K)
PC	Critical pressure (bar)
VC	Critical volume (m3)
OMEGA	Acentric factor
PARLASSC	Parameter for LASSC equation of state
DHVAP	Vaporisation enthalpy at TB
PVAP	Boiling curve coefficients

PROPERTY	DESCRIPTION
CPL	Liquid heat capacity coefficients
CPID	Ideal gas heat capacity coefficients
VSAT	Molar volume at reference temperature TREF

TABLE 3 - DATA NEEDED FOR ENTHALPY AND ENTROPY CALCULATIONS OF LIQUIDS AND VAPOURS

VOLUME CALCULATIONS

Table below presents, for each listed calculation method, the set of fundamental properties whose values must be present in the model file for each chemical compound of the mixture.

	P	G A	
	A	Y S	
	R	D R S D	
	O L	I A O 6	
	M A V P D C 0		
	E S S O I P /		
T P V G S A L U A 6			
C C C A C T M S R 0			
 VIDEAL			
VIRIAL	X X		X X X
GUNN-YAMADA	X X X X	X	
PENG-ROBINSON	X X	X	
LASSC	X X	X X	
SOAVE	X X	X	
LEE-KESLER	X X X X		
CURL-PITZER	X X	X	
WUITHIER COLDDENS			X

PROPERTY	DESCRIPTION
TC	Critical temperature (K)
PC	Critical pressure (bar)

PROPERTY	DESCRIPTION
VC	Critical volume (m3)
OMEGA	Acentric factor
PARLASSC	Parameter for LASSC equation of state
VSAT	Molar volume at reference temperature TREF
DIPOLM	Dipole moment
GYRADIUS	Gyration radius
ASSOCPAR	Associative parameter
D60/60	Density 60/60

TABLE 4 - DATA NEEDED FOR VOLUME CALCULATIONS OF LIQUIDS AND VAPOURS

VISCOSITY AND THERMAL CONDUCTIVITY CALCULATIONS

Table below presents, for each listed calculation method, the set of fundamental properties, whose values must be present in the model file for each chemical compound of the mixture.

	D O D I C M C H P V V O O E P V O S I N L T T P G I A L A S D W B C C A D P M T L L												
PEDERSEN	X	X											
VANVELZEN												X	
LETSOU-STIEL	X	X	X	X									
STIEL-THODOS	X	X			X								
POLYNOM													X
ROBBINS-KINGREA	X	X			X	X							
CHAPMAN-ENSKOG	X	X					X	X					

PROPERTY	DESCRIPTION
MOLW	Molecular weight
TB	Normal boiling temperature (K)
TC	Critical temperature (K)
PC	Critical pressure (bar)
OMEGA	Acentric factor
CPID	Ideal gas heat capacity coefficients
DHVAP	Vaporisation enthalpy at TB
DIPOLM	Dipole moment
VSAT	Molar volume at reference temperature TREF
VISL	Liquid viscosity coefficients
CONDL	Liquid thermal conductivity coefficients

TABLE 5 - DATA NEEDED FOR VISCOSITY AND THERMAL CONDUCTIVITY CALCULATIONS OF LIQUIDS AND VAPOURS

SECTION 4: INTERACTION PARAMETERS

Many calculation methods contain binary interaction parameters that can be specified. When not specified, default values are used instead. These interaction parameters may be found in the literature or may be obtained by parameter identification.

The parameter identification is an iterative procedure as follows :

- calculate the phase compositions for given values of interaction parameters
- compare them to experimental data
- calculate new parameter values, which reduce the deviations

A separate program allows you to estimate these parameters.

In state equations, identical parameters are used for both phases.

When state equations are used to calculate vapour or liquid enthalpies and entropies, VALI uses the default values of the interaction parameters.

Binary parameters can be entered and edited in the Binary Parameters page of the Thermod Notebook.

The screenshot shows the 'Thermod: THER1' window with the 'Binary Parameters' tab selected. The window contains a form for adding binary parameters and a table of existing parameters.

Binary parameter form:

- Compound: C3
- Method: Soave
- Compound: C3-
- Parameter: DELTA
- Add button
- Delete button
- Value: 0

Table of existing parameters:

Substance	Substance	Method	Parameter	Value
C3	C3-	Peng-Robinson	DELTA	0.1023

At the bottom of the window are buttons for OK, Cancel, and Help.

FIGURE 3 - THE BINARY PARAMETERS PAGE OF A THERMOD NOTEBOOK

4.1 EQUATIONS OF STATE

There is one set of interaction parameters for each pair of chemical compounds present in the mixture. These parameters are involved in the mixing rules used to calculate the mixture's equation of state coefficients.

Table below gives their names and their default values in the BELSIM environment.

CALCULATION METHOD	NAME	DEFAULT VALUE
SOAVE	DELTA	0.0
PENG-ROBINSON	DELTA	0.0
LEE-KESLER	K	1.0
LASSC	DELTA	0.0
	TAU	1.0

TABLE 6 - INTERACTION PARAMETERS FOR EQUATIONS OF STATE

4.2 VIRIAL EQUATION OF STATE

The calculation of the second virial coefficients of the pure chemical compounds considers the effects of the dipole moment and of the associative character of the chemical compounds.

When the association parameter is higher than 4.5 (as with carboxylic acids), the chemical compounds are found as monomers and dimers. This effect is automatically considered in the calculation of the virial coefficients.

Interactions between different chemical compounds are calculated from the solvation parameters, which can be identified by a separate program.

Table below shows the name and the default values of the VIRIAL interaction parameters.

CALCULATION METHOD	NAME	DEFAULT VALUE
VIRIAL	ETA	0.0

TABLE 7 - VIRIAL EQUATION, INTERACTION PARAMETERS

4.3 NRTL AND UNIQUAC METHODS

These equations calculate the activity coefficients of liquid chemical compounds in non-ideal mixtures.

The NRTL model contains six parameters: $ijC0$ $ijCT$ $jiC0$ $jiCT$ $ijA0$ $ijAT$

These are involved in the relationships:

$$(g_{ij} - g_{jj}) / R = ijC0 + ijCT * T^{-K}$$

$$(g_{ji} - g_{ii}) / R = jiC0 + jiCT * T^{-K}$$

$$A_{ij} = ijA0 + ijAT * T$$

Where:

- g_{ij} , g_{ji} , g_{ii} , g_{jj} and A_{ij} are the coefficients from the NRTL equation for the free energy and R is the perfect gas constant

The UNIQUAC model contains four parameters: A_{ij} B_{ij} A_{ji} B_{ji} .

These are involved in the relationships:

$$(U_{ij} - U_{jj}) / R = A_{ij} + B_{ij} / T$$

$$(U_{ji} - U_{ii}) / R = A_{ji} + B_{ji} / T$$

Where:

- U_{ij} , U_{jj} , U_{ji} and U_{ii} are the coefficients from the UNIQUAC equation for free energy

When supercritical compounds are present in the mixture, you must provide the coefficients to calculate their activity at infinite dilution:

$$\ln \gamma_{am} = isGAM0 + isGAMT / T.$$

Table below shows the names of these interaction parameters in the BELSIM environment.

CALCULATION METHOD	NAME	DEFAULT VALUE
NRTL	$ijA0$	0.0
	$ijC0$	0.0
	$jiC0$	0.0
	$ijAT$	0.0
	$ijCT$	0.0
	$jiCT$	0.0

CALCULATION METHOD	NAME	DEFAULT VALUE
	isGAM0	0.0
	isGAMT	0.0
UNIQUAC	Aij	0.0
	Aji	0.0
	Bij	0.0
	Bji	0.0
	isGAM0	0.0
	isGAMT	0.0

TABLE 8 - NRTL AND UNIQUAC INTERACTION PARAMETERS

4.4 QUADRATIC HEAT OF MIXING

QUADRAT1 and QUADRAT2 calculates a deviation from respectively LIDEAL and LIDEAL2 methods by adding a quadratic mixing enthalpy calculated as:

$$DH_{mix} = \sum ijENTH \cdot x_i \cdot x_j$$

Where:

- $ijENTH$ is the binary parameter, expressed in kJ/kmol
- x_i and x_j are the molar fractions of the chemical compounds i and j in the liquid phase

CALCULATION METHOD	NAME	DEFAULT VALUE
QUADRAT1/QUADRAT2	ijENTH	0.0

TABLE 9 - QUADRATIC MIXING ENTHALPY, INTERACTION PARAMETERS

SECTION 5: CALCULATION METHODS FOR TYPICAL MIXTURES

This section, shows several types of mixtures and proposes methods to calculate their properties.

The types of mixtures which will be analysed are :

- pure water
- nonpolar hydrocarbon mixtures with or without water
- oxygenated compounds mixtures

5.1 WATER

The Thermod STEAM present in Emty.bls file is recommended.

Use the IAPWS-IF97, NBS or VDI methods for all the pure water. Note that IAPWS-IF97 and VDI methods must be used for all properties together. You cannot combine IAPWS-IF97, VDI and other methods.

Note also that you in most units you cannot mix streams using IAPWS-IF97 or VDI methods with other streams, because the reference conditions for enthalpy are different. Only BBXVAL units enable to mix streams using IAPWS-IF97 or VDI methods.

5.2 NONPOLAR HYDROCARBON MIXTURES WITH OR WITHOUT WATER

TYPICAL THERMODS

The file Empty.bls contains the following Thermods:

- Gas: natural gas: N2, H2, H2S, CO2, C1, C2, C3, I-C4, N-C4, N-C5
- Fuel: fuel-oil: C_FUEL, H_FUEL, S_FUEL, H2O_FUEL
- Coal: coal: C_COAL, H_COAL, S_COAL, H2O_COAL
- Air: air: O2, N2
- Fumes: fumes: O2, N2, CO2, H2O, SO2

The models use Ideal methods and are recommended for Power Plants applications.

The file Refining.bls contains in addition:

- SOAVE: Empty thermod based on SOAVE methods
- P-R: Empty thermod based on Peng-robinson methods
- CRUDE: Typical thermod to be used to model Crudes and petroleum cuts

You can derive your own models by using these models as starting point.

If water is present in the mixture and could form a separate water liquid phase, use one of the following strings for the Stream State parameter :

- Water/Liquid/Vapour
- Water/Liquid

In other cases, use one of the following strings for this parameter :

- Liquid
- Vapour
- Liquid/Vapour

CALCULATION OF FUGACITIES

At low and medium pressures, WITHOUT light chemical compounds, choose between the following:

- VIDEAL and LIDEAL calculation methods for vapour and liquid phases
- REDLICH-KWONG for vapour phase and CHAO-SEADER for liquid phase
- VIDEAL for vapour phase and BRAUN K10 for the liquid phase for mixtures of heavy hydrocarbon compounds
- any of the equations of state : SOAVE, PENG-ROBINSON, LASSC

With the first three pairs of methods, there are no interaction parameters to provide.

For the equations of state, use the default values for the interaction parameters.

When light compounds are present in the mixture, choose any of the equations of state, but in this case, we advise you to identify the interaction parameters between the *light compounds* and the heavier compounds using experimental data of solubilities. A separate program can help you to perform this identification.

If the light compounds are H₂ and CH₄, the methods of REDLICH-KWONG and GRAYSON-STREED may be respectively used for the vapour and liquid phases.

CALCULATION OF ENTHALPIES AND ENTROPIES

Several methods are available to calculate the enthalpy and entropy of nonpolar hydrocarbon mixtures.

Choose among the following :

- LIDEAL2 for the liquid phase
- VIDEAL for the vapour phase
- CURL-PITZER for both phases
- LEE-KESLER for both phases

The pressure and mixing corrections on enthalpy and entropy are only calculated in the CURL-

PITZER and LEE-KESLER methods.

CALCULATION OF MOLAR VOLUMES

Choose among the following :

- GUNN-YAMADA for the liquid phase
- Whuithier for the liquid phase of hydrocarbon mixtures
- VIDEAL for the vapour phase
- CURL-PITZER for both phases
- any of the equations of state : SOAVE, PENG-ROBINSON, LASSC

with a preference for the LASSC equation when interaction parameters are available. Volume predictions by equations of state are rarely accurate.

CALCULATION OF THERMAL CONDUCTIVITIES AND VISCOSITIES

For thermal conductivities and viscosities, use the PEDERSEN method for both phases.

5.3 OXYGENATED COMPOUND MIXTURES

Select one of the following strings for the Stream State parameter :

- Liquid
- Vapor
- Liquid-Vapor
- Liquid-Liquid
- Liquid-Liquid-Vapor

CALCULATION OF FUGACITIES

At low and medium pressures, even for polar chemical compounds, choose between the following:

- VIDEAL and LIDEAL for vapour and liquid phases
- VIRIAL and NRTL for the vapour and liquid phases
- VIRIAL and UNIQUAC for the vapour and liquid phases
- VIRIAL and UNIFAC for the vapour and liquid phases

For the second and third methods, interaction parameters for both phases have to be identified from liquid-vapour equilibrium data. A separate program may help you to identify these parameters.

The last method is equivalent to UNIQUAC, but interaction parameters are defined from structure of each molecule : it is based on group contributions (list of groups is defined with pure component

properties).

When supercritical compounds are present in the mixture, you must provide their activity coefficient at infinite dilution.

CALCULATION OF ENTHALPIES AND ENTROPIES

Several methods are available for the calculation of enthalpies and entropies of vapour and liquid. You can choose from amongst the following methods :

- LIDEAL2 for the liquid phase
- VIDEAL for the vapour phase
- LEE-KESLER equation of state for both phases

Neither mixing enthalpies, nor pressure corrections can be taken into account, except with the method of LEE-KESLER.

CALCULATION OF MOLAR VOLUMES

For the calculation of molar volumes of vapour and liquid phases, several methods are available :

- GUNN-YAMADA for the liquid phase
- VIDEAL for the vapour phase
- any of the equations of state : SOAVE, PENG-ROBINSON, LASSC

with a preference for the LASSC equation when the interaction parameters are available. Volume predictions by equations of state are generally poor.

CALCULATION OF THERMAL CONDUCTIVITIES

For the calculation of thermal conductivities, the following methods may be used :

- STIEL-THODOS for the vapour phase
- ROBBINS-KINGREA for the liquid phase

CALCULATION OF VISCOSITIES

For the calculation of viscosities, the following methods are available :

- CHAPMAN-ENSKOG for the vapour phase
- VAN-VELZEN for the liquid phase

SECTION 6: REFERENCES

Statistical Thermodynamics of Liquid Mixtures : A New Expression for the Excess Gibbs Energy of Partly or Completely Miscible Systems

Abrams D.S. and Prausnitz J.

AIChEJ 116, Vol 21, No 1, 1975.

A General Correlation of Vapour-liquid Equilibria in Hydrocarbon Mixtures

Chao K. C. and Seader J. D.

AIChE Journal, pg. 598, Dec. 1961.

Vapour liquid equilibria of high temperature, high pressure systems

Grayson H. G. and Streed G. W.

6th world petroleum congress, West Germany, June 1963.

NBS/NRC Steam tables, thermodynamic and transport properties and computer programs for vapour and liquid states of water in SI units

Haar L., Gallagher J.S. and Kell G.S.,

Hemisphere Publishing Corporation (1984)

Equations for the viscosity and thermal conductivity coefficient of Methane

Hanley H.J.M., McCarty R.D., Haynes W.M.

Cryogenics p413 (1975)

A Generalized Method for Predicting Second Virial Coefficients

Hayden J.G. et O'Connel J.P.

Ind. Eng. Chem., Proc. Des. Dev. 14(1) : 209 (1975)

Contribution à l'étude de la thermodynamique des systèmes non idéaux sous pressions élevées

Heyen G.

Thèse , Université de Liège, Belgique. (1982)

A modified Benedict-Webb-Rubin Equation of State for Methane using recent experimental Data

McCarty R.D.

Cryogenics p277 (1974)

Generalized Correlation for Fugacity Coefficients in Mixtures at Moderate Pressures. Application of Chemical Theory of Vapour Imperfections

Nothnagel K-H, Abrams D.S. et Prausnitz J.M.

Ind. Eng. Chem., Proc. Des. Dev. 12(1) : 25 (1973)

Prediction of transport and Surface Properties

Pedersen K.S., Fredenslund A.

Chem. Eng. Fund., XVIII Congress, 1987, p 299.1

Viscosity of Crude Oils

Pedersen K.S., Fredenslund A., Christensen P.L., Thomassen P.

Chem. Eng. Sci, Vol.39, N°6, p1101 (1984)

A Corresponding States Model for the Thermal Conductivity of Gases and Liquid

Pedersen K.S., Fredenslund A.

Chem. Eng. Sci, Vol.35, p871 (1980)

An Improved Corresponding States Model for the Prediction of Oil and Gas Viscosities and thermal Conductivity

Pedersen K.S., Fredenslund A.

Chem. Eng. Sci. Vol.42, N°1; p182 (1987)

A new Two-Constant Equation of State

Peng D.Y. et Robinson D.B.

Ind. Eng. Chem., Fundamental 15(1) : 59-64 (1976)

Calculation of high-pressure vapour-liquid equilibria from a corresponding states correlation with emphasis on asymmetric mixtures

Plocker U., Knapp H, Prausnitz J.,

Ind. Eng. Chem., Proc. Des. Dev. 324, vol.17, n3, 1978.

The properties of gases and liquids

Reid R. C., Prausnitz J.M. et Sherwood T.K.

Third Edition 1977 McGRAW-HILL BOOK COMPANY

Calcul sur ordinateur des équilibres liquide-vapeur et liquide-liquide

Renon H., Asselineau L., Cohen G., Raimbault C.

1971, Technip.

Equilibrium constants from a modified Redlich-Kwong equation of state

Soave G.

Chemical Engineering Science, 27:1197 (1972)

Méthodes appliquées aux raffinages

Vidal J.

Publications de l'Institut Français du Pétrole,

Editions technip, 1974

Properties of Water and Steam- The Industrial Standard IAPWS-IF97 for the Thermodynamic Properties and Supplementary Equations for Other Properties

Wagner W., Kruse A.,

Springer-Verlag, Berlin 1998

Curl R.F., K.S. Pitzer K. S.

Ind. Eng. Chem., 50,265 (1958)

Cajander B.C., Hipkin H.G., Lenoir J.M.,

J. Chem. Engr. Data, 5, 251 (1960)

SECTION 7: FLUIDS DATA BASE

The FLUIDS compounds are gathered in families:

Acetylenes, Acids, Alcohols, Aldehydes, Amides, Aromatics, Esters, Ethers, Halogenated derivatives, Isocyanates, Ketones, Miscellaneous, Naphtenes, Nitrogen derivatives, Olefins, Oxides, Paraffins, Sulfur derivatives.

The properties stored for each compound are listed in the following table.

PARAMETER	DESCRIPTION	PHYSICAL UNITS
H0F	Standard enthalpy of formation	kJ/kmol
G0F	Gibbs s std. energy of formation	kJ/kmol
MOLW	Molecular weight	kg/kmol
TB	Boiling temperature	K
TC	Critical temperature	K
PC	Critical pressure	bar
VC	Critical volume	m ³ /kmol
OMEGA	Pitzer's acentric factor	-
CPLA, CPLB, CPLC, CPLD	Liquid Heat Capacity CPL = A + B*T + C*T ² + D*T ³	kJ/kmol/K
TMIN(PV)	Temperature range for vapour pressure	K
TMAX(PV)		
PVAPA, PVAPB, PVAPC, PVAPD, PVAPE, PVAPF, PVAPG, PVAPH	Vapour pressure $\ln(PVAP) = A + B/(T+C) + D*T + E*T^2 + F*\ln(T) + G.TH$	bar
CPIDA, CPIDB, CPIDC, CPIDD	Ideal gas heat capacity CPID = A + B*T + C*T ² + D*T ³	kJ/kmol/K
DHVP(B)	Vaporisation enthalpy at TEB	kJ/kmol
DIPOLM	Dipolar moment	Debyes

PARAMETER	DESCRIPTION	PHYSICAL UNITS
VMOLREF	Saturated liquid molar volume TREF	m ³ /kmol
VISL1, VISL2	Liquid viscosity $\log(\text{VISL}) = \text{VISL1} * (1/T - 1/\text{VISL2})$	N.s/m ²
EN.CARAC	Characteristic energy	K
COLDIAM	Collision diameter	m
THCNDPAR	Thermal conductivity parameter (Robbins-Kingrea)	
GYRADIUS	Gyration radius	m
ASSOCPAR	Association parameter	-
CONDL1, CONDL2, CONDL3	Liquid conductivity $\text{CONDL} = \text{CONDL1} + \text{CONDL2} * T + \text{CONDL3} * T^2$	kJ/m/K
TMELT	Melting temperature	K
DHMELT	Melting enthalpy	kJ/kmol
SURFTEN1, SURFTEN2	Othmer's surface tension $\text{SURFTEN} = \text{SURFTEN1} * (T_c - T) ** \text{SURFTEN2}$	N/m
TREF(VM)	Reference temperature for VMOLREF	K
TMIN(CL) TMAX(CL)	Temperature range of CPL	K
TMIN(CG) TMAX(CG)	Temperature range of CPID	K
KLASSC, MLASSC, TETLASSC, NLASSC, NHTLASSC	PARAMETERS FOR LASSC'S EQUATION OF STATE	
RVDW	Van der Waals volume	-

KLASSC, MLASSC, TETLASSC, NLASSC, NHTLASSC	PARAMETERS FOR LASSC'S EQUATION OF STATE	
QVDW	Van der Waals surface area	-
PARSOLUB	Hildebrand solubility parameter	(kJ/m ³) ^{1/2}
D60/60	Density	60/60
UNUSED57		
UNUSED58		
UNUSED59		
UNUSED60		
LHV	Lower Heating Value	kJ/kmol
UHV	Upper heating Value	kJ/kmol
UNUSED63		
UNUSED64		
UNUSED65		
UNUSED66		
UNUSED67		
UNUSED68		
UNUSED69		
UNUSED70		

The list of compounds available in this library is shown below.

ACETYLENES

LONG NAME	FORMULA	SHORT NAME
Acetylene	C ₂ H ₂	C ₂ =
Propyne	C ₃ H ₄	C ₃ =

LONG NAME	FORMULA	SHORT NAME
1-Butyne	C4H6	1C4=
2-Butyne	C4H6	2C4=
Butadiyne	C4H2	C4==
1-Pentyne	C5H8	C5=

ACIDS

LONG NAME	FORMULA	SHORT NAME
Acetic Acid	C2H4O2	HAC
Acrylic Acid	C3H4O2	HACRYL
Benzoic Acid	C7H6O2	HBZ
Formic Acid	C1H2O2	HFORMIC
I-Butyric Acid	C4H8O2	I-HBUTYR
N-Butyric Acid	C4H8O2	N-HBUTYR
Nitric Acid	H1O3N1	HNO3
N-Valeric Acid	C5H10O2	N-HVALER
Propionic Acid	C3H6O2	HPROPIO
Succinic Acid	C4H6O4	HSUCCIN
Sulfuric Acid	H2O4S1	H2SO4
Trifluoroacetic Acid	C2H1O2F3	HTFAC

ALCOOLS

LONG NAME	FORMULA	SHORT NAME
Methanol	C1H4O1	C1OL
Ethanol	C2H6O1	C2OL
Ethylene Glycol	C2H6O2	C2-2OL
Allyl Alcohol	C3H6O1	ALLOL
1-Propanol	C3H8O1	1C3OL
2-Propanol	C3H8O1	2C3OL
12-Propanediol	C3H8O2	12C32OL
13-Propanediol	C3H8O2	13C32OL
Glycerol	C3H8O3	GLYCEROL
N-Butanol	C4H10O1	NC4OL

LONG NAME	FORMULA	SHORT NAME
2-Butanol	C4H10O1	2C4OL
I-Butanol	C4H10O1	IC4OL
Tert-Butanol	C4H10O1	TC4OL
1,3-Butanediol	C4H10O2	12C4OL
2,3-Butanediol	C4H10O2	23C4OL
1,2,3-Butanetriol	C4H10O3	123C4OL
1-Pentanol	C5H12O1	1C5OL
2-Methyl-1-Butanol	C5H12O1	2M1C4OL
3-Methyl-1-Butanol	C5H12O1	3M1C4OL
2-Methyl-2-Butanol	C5H12O1	2M2C4OL
22-Dimethyl-1-Propanol	C5H12O1	22DMPOL
2-Pentanol	C5H12O1	2C5OL
3-Pentanol	C5H12O1	3C5OL
3-Methyl-2-Butanol	C5H12O1	3M2C4OL
1,5-Pentanediol	C5H12O2	15C5OL
3-Methyl-1,3-Butanediol	C5H12O2	3M13C4OL
2,3,4-Pentanetriol	C5H12O3	234C5OL
Cyhexanol	C6H12O1	CYC6OL
Phenol	C6H6O1	PHENOL
13-Dihydroxybenzene	C6H6O2	13DHB
14-Dihydroxybenzene	C6H6O2	14DHB
12-Dihydroxybenzene	C6H6O2	12DHBZ
1-Hexanol	C6H14O1	1C6OL
2-Hexanol	C6H14O1	2C6OL
3-Hexanol	C6H14O1	3C6OL
2-Methyl-1-Pentanol	C6H14O1	2M1C5OL
3-Methyl-1-Pentanol	C6H14O1	3M1C5OL
4-Methyl-1-Pentanol	C6H14O1	4M1C5OL
2-Methyl-2-Pentanol	C6H14O1	2M2C5OL
3-Methyl-2-Pentanol	C6H14O1	3M2C5OL
4-Methyl-2-Pentanol	C6H14O1	4M2C5OL
2-Methyl-3-Pentanol	C6H14O1	2M3C5OL
3-Methyl-3-Pentanol	C6H14O1	3M3C5OL

LONG NAME	FORMULA	SHORT NAME
2-Ethyl-1-Butanol	C ₆ H ₁₄ O ₁	2E1C4OL
2,2-Dimethyl-1-Butanol	C ₆ H ₁₄ O ₁	22M1C4OL
2,3-Dimethyl-2-Butanol	C ₆ H ₁₄ O ₁	23M1C4OL
3,3-Dimethyl-2-Butanol	C ₆ H ₁₄ O ₁	33M1C4OL
2-Methyl-2,4-Pentanediol	C ₆ H ₁₄ O ₂	2M24C5OL
1,2,6-Hexanetriol	C ₆ H ₁₄ O ₃	126C6OL
Benzylalcohol	C ₇ H ₈ O ₁	BZOL
O-Cresol	C ₇ H ₈ O ₁	O-CROL
M-Cresol	C ₇ H ₈ O ₁	M-CROL
P-Cresol	C ₇ H ₈ O ₁	P-CROL
Guaiacol	C ₇ H ₈ O ₂	GUAIACOL
1-Heptanol	C ₇ H ₁₆ O ₁	1C7OL
2-Heptanol	C ₇ H ₁₆ O ₁	2C7OL
3-Heptanol	C ₇ H ₁₆ O ₁	3C7OL
4-Heptanol	C ₇ H ₁₆ O ₁	4C7OL
2-Methyl-2-Hexanol	C ₇ H ₁₆ O ₁	2M2C6OL
2-Methyl-3-Hexanol	C ₇ H ₁₆ O ₁	2M3C6OL
3-Methyl-3-Hexanol	C ₇ H ₁₆ O ₁	3M3C6OL
3-Ethyl-3-Pentanol	C ₇ H ₁₆ O ₁	3E3C5OL
2,3-Dimethyl-3-Pentanol	C ₇ H ₁₆ O ₁	23M3C5OL
2,4-Dimethyl-3-Pentanol	C ₇ H ₁₆ O ₁	24M3C5OL
2-Benzofuranol	C ₈ H ₆ O ₂	BZFROL
O-Ethylphenol	C ₈ H ₁₀ O ₁	O-EPHOL
M-Ethylphenol	C ₈ H ₁₀ O ₁	M-EPHOL
P-Ethylphenol	C ₈ H ₁₀ O ₁	P-EPHOL
Phenetole	C ₈ H ₁₀ O ₁	PHNTOL
23-Xylenol	C ₈ H ₁₀ O ₁	23XYOL
24-Xylenol	C ₈ H ₁₀ O ₁	24XYOL
25-Xylenol	C ₈ H ₁₀ O ₁	25XYOL
26-Xylenol	C ₈ H ₁₀ O ₁	26XYOL
34-Xylenol	C ₈ H ₁₀ O ₁	34XYOL
35-Xylenol	C ₈ H ₁₀ O ₁	35XYOL
2-Phenylethylalcohol	C ₈ H ₁₀ O ₁	2PHEOL

LONG NAME	FORMULA	SHORT NAME
4-Methylguaiacol	C ₈ H ₁₀ O ₂	4MGCOL
1-Octanol	C ₈ H ₁₈ O ₁	1C8OL
2-Octanol	C ₈ H ₁₈ O ₁	2C8OL
2-Ethylhexanol	C ₈ H ₁₈ O ₁	2EC6OL
3-Octanol	C ₈ H ₁₈ O ₁	3C8OL
4-Octanol	C ₈ H ₁₈ O ₁	4C8OL
2-Methyl-1-Heptanol	C ₈ H ₁₈ O ₁	2M1C7OL
3-Methyl-1-Heptanol	C ₈ H ₁₈ O ₁	3M1C7OL
4-Methyl-1-Heptanol	C ₈ H ₁₈ O ₁	4M1C7OL
5-Methyl-1-Heptanol	C ₈ H ₁₈ O ₁	5M1C7OL
6-Methyl-1-Heptanol	C ₈ H ₁₈ O ₁	6M1C7OL
2-Methyl-2-Heptanol	C ₈ H ₁₈ O ₁	2M2C7OL
3-Methyl-2-Heptanol	C ₈ H ₁₈ O ₁	3M2C7OL
4-Methyl-2-Heptanol	C ₈ H ₁₈ O ₁	4M2C7OL
4-Indanol	C ₉ H ₁₀ O ₁	4INDANOL
5-Indanol	C ₉ H ₁₀ O ₁	5INDANOL
1-Indanol	C ₉ H ₁₀ O ₁	1INDANOL
2-Indanol	C ₉ H ₁₀ O ₁	2INDANOL
N-Propylphenol	C ₉ H ₁₂ O ₁	NPPHENOL
N-Nonanol	C ₉ H ₂₀ O ₁	NC9OL
1-Napthol	C ₁₀ H ₈ O ₁	1NPHTOL
2-Napthol	C ₁₀ H ₈ O ₁	2NPHTOL
P-Tert-Butyphenol	C ₁₀ H ₁₄ O ₁	BPHNOL
P-Ethyl-B-Phenylethanol	C ₁₀ H ₁₄ O ₁	P-EPEOL
O-Ethyl-B-Phenylethanol	C ₁₀ H ₁₄ O ₁	O-EPEOL
1-Decanol	C ₁₀ H ₂₂ O ₁	1C10OL
1-Dodecanol	C ₁₂ H ₂₆ O ₁	1C12OL
Heptadecanol	C ₁₇ H ₃₆ O ₁	C17OL
1-Octadecanol	C ₁₈ H ₃₈ O ₁	1C18OL
1-Eicosanol	C ₂₀ H ₄₂ O ₁	1C20OL

ALDEHYDES

LONG NAME	FORMULA	SHORT NAME
Formaldehyde	C1H2O1	C1AL
Acetaldehyde	C2H4O1	C2AL
Propionaldehyde	C3H6O1	C3AL
Methylal	C3H8O2	METHYLAL
N-Butanal	C4H8O1	N-BUTAL
I-Butanal	C4H8O1	I-BUTAL
Furfural	C5H4O2	FURFURAL
Valeraldehyde	C5H10O1	VALERAL
Benzaldehyde	C7H6O1	BZAL
N-Hexanal	C6H12O1	N-C6AL
N-Heptanal	C7H14O1	N-C7AL
N-Octanal	C8H16O1	N-C8AL
N-Nonanal	C9H18O1	N-C9AL
N-Decanal	C10H20O1	N-C10AL

AMIDES

LONG NAME	FORMULA	SHORT NAME
Methylamine	C1H5N1	MA
Methylhydrazine	C1H6N2	C1-HZINE
Ethylamine	C2H7N1	EA
Dimethylamine	C2H7N1	DMA
Monoethanolamine	C2H7O1N1	MEA
Ethylenediamine	C2H8N2	C2-DA
Dimethylformamide	C3H7O1N1	DMF
N-Propylamine	C3H9N1	N-PA
I-Propylamine	C3H9N1	I-PA
Trimethylamine	C3H9N1	TMA
Morpholine	C4H9O1N1	MORPHLIN
N-Butylamine	C4H11N1	N-BA
I-Butylamine	C4H11N1	I-BA
Diethylamine	C4H11N1	DEA
Pyridine	C5H5N1	PYRIDINE
Aniline	C6H7N1	ANILINE

LONG NAME	FORMULA	SHORT NAME
4-Methylpyridine	C6H7N1	4MPYR
2-Methylpyridine	C6H7N1	2MPYR
3-Methylpyridine	C6H7N1	3MPYR
13-Benzenediamine	C6H8N2	13BZDA
2-Methyl-Piperidine	C6H13N1	2MPIPER
Dipropylamine	C6H15N1	DPA
Triethylamine	C6H15N1	TEA
23-Dimethylpyridine	C7H9N1	23DMPYR
25-Dimethylpyridine	C7H9N1	25DMPYR
34-Dimethylpyridine	C7H9N1	34DMPYR
35-Dimethylpyridine	C7H9N1	35DMPYR
24-Dimethylpyridine	C7H9N1	24MPYR
26-Methylpyridine	C7H9N1	26MPYR
2-Ethylpyridine	C7H9N1	2EPYR
3-Ethylpyridine	C7H9N1	3EPYR
4-Ethylpyridine	C7H9N1	4EPYR
Methylphenylamine	C7H9N1	MPHA
O-Toluidine	C7H9N1	O-TOLUID
M-Toluidine	C7H9N1	M-TOLUID
P-Toluidine	C7H9N1	P-TOLUID
67-Dihydro-5h-1-Pyridine	C8H9N1	671PYRID
67-Dihydro-5h-2-Pyridine	C8H9N1	672PYRID
N-Ethylaniline	C8H11N1	N-EANILI
236-Trimethylpyridine	C8H11N1	236MPYR
246-Trimethylpyridine	C8H11N1	246MPYR
2-Methyl-5-Ethylpyridine	C8H11N1	2M5EPYR
24-Dimethylaniline	C8H11N1	24MANILI
25-Dimethylaniline	C8H11N1	25MANILI
26-Dimethylaniline	C8H11N1	26MANILI
2-Ethylaniline	C8H11N1	2EANILIN
23-Dimethylaniline	C8H11N1	23DMANIL
35-Dimethylaniline	C8H11N1	35DMANIL
4-Ethylaniline	C8H11N1	4EANILIN

LONG NAME	FORMULA	SHORT NAME
Nn-Dimethylaniline	C8H11N1	NN-DMA
2-C1-5-C2-Piperidine	C8H17N1	2M5EPIPE
Dibutylamine	C8H19N1	DBA
1-Naphthylamine	C10H9N1	1NPHA
N-Butylaniline	C10H15N1	N-BANILI
2-Phenylpyridine	C11H9N1	2PHPYR
Tributylamine	C12H27N1	TBA

AROMATICS

LONG NAME	FORMULA	SHORT NAME
Benzene	C6H6	BZ
Toluene	C7H8	TOLUENE
Styrene	C8H8	STYRENE
O-Xylene	C8H10	O-XYL
M-Xylene	C8H10	M-XYL
P-Xylene	C8H10	P-XYL
Ethylbenzene	C8H10	EBZ
Indene	C9H8	INDENE
Indan	C9H10	INDAN
Alpha-Methylstyrene	C9H10	AMSTY
M-Methylstyrene	C9H10	MMSTYR
O-Methylstyrene	C9H10	OMSTYR
P-Methylstyrene	C9H10	PMSTYR
N-Propylbenzene	C9H12	NPBZ
I-Propylbenzene	C9H12	IPBZ
1-Methyl-2-Ethylbenzene	C9H12	1M2EBZ
1-Methyl-3-Ethylbenzene	C9H12	1M3EBZ
1-Methyl-4-Ethylbenzene	C9H12	1M4EBZ
123-Trimethylbenzene	C9H12	123TMBZ
124-Trimethylbenzene	C9H12	124TMBZ
135-Trimethylbenzene	C9H12	135TMBZ
Naphthalene	C10H8	NAPHT
1-Methylindene	C10H10	1MINDENE

LONG NAME	FORMULA	SHORT NAME
2-Methylindene	C10H10	2MINDENE
3-Methylindene	C10H10	3MINDENE
4-Methylindene	C10H10	4MINDENE
5-Methylindene	C10H10	5MINDENE
6-Methylindene	C10H10	6MINDENE
7-Methylindene	C10H10	7MINDENE
1,2-Dihydronaphthalene	C10H10	DHNAPHTA
1234-Tetrahydronaphthalen	C10H12	1234HNPH
Dicyclopentadiene	C10H12	DCYC5=
1-Methylindan	C10H12	1MINDAN
N-Butylbenzene	C10H14	N-BBZ
I-Butylbenzene	C10H14	I-BBZ
Sec-Butylbenzene	C10H14	S-BBZ
Tert-Butylbenzene	C10H14	T-BBZ
1-Methyl-2-I-Propylbenzn	C10H14	1M2PBZ
1-Methyl-3-I-Propylbenzn	C10H14	1M3PBZ
1-Methyl-4-I-Propylbenzn	C10H14	1M4PBZ
14-Diethylbenzene	C10H14	14DEBZ
1245-Tetramethylbenzene	C10H14	1245BZ
1234-Tetramethylbenzene	C10H14	1234TMBZ
1235-Tetramethylbenzene	C10H14	1235TMBZ
2-Ethyl-M-Xylene	C10H14	2EMXYL
2-Ethyl-P-Xylene	C10H14	2EPXYL
3-Ethyl-O-Xylene	C10H14	3EOXYL
4-Ethyl-O-Xylene	C10H14	4EOXYL
4-Ethyl-M-Xylene	C10H14	4EMXYL
5-Ethyl-M-Xylene	C10H14	5EMXYL
12-Diethylbenzene	C10H14	12DEB
13-Diethylbenzene	C10H14	13DEB
O-Propyltoluene	C10H14	OPTOLU
M-Propyltoluene	C10H14	MPTOLU
P-Propyltoluene	C10H14	PPTOLU
1-Methylnaphthalene	C11H10	1MNAPHT

LONG NAME	FORMULA	SHORT NAME
2-Methylnaphthalene	C ₁₁ H ₁₀	2MNAPHT
23-Dimethylindan	C ₁₁ H ₁₂	23MINDAN
16-Dimethylindan	C ₁₁ H ₁₄	16MINDAN
47-Dimethylindan	C ₁₁ H ₁₄	47MINDAN
2-Methyl-Tetralin	C ₁₁ H ₁₄	2MTTLIN
6-Methyl-Tetralin	C ₁₁ H ₁₄	6MTTLIN
5-C2-124-Tric1-Benzene	C ₁₁ H ₁₆	5E124MBZ
2-C2-135-Tric1-Benzene	C ₁₁ H ₁₆	2E135MBZ
5-Propyl-M-Xylene	C ₁₁ H ₁₆	5PMXYL
Pentamethylbenzene	C ₁₁ H ₁₆	5MBZ
3-Propyl-O-Xylene	C ₁₁ H ₁₆	3POXYL
2-Propyl-M-Xylene	C ₁₁ H ₁₆	2PMXYL
2-Propyl-P-Xylene	C ₁₁ H ₁₆	2PPXYL
23-Dimethylcumene	C ₁₁ H ₁₆	23MCUM
35-Dimethylcumene	C ₁₁ H ₁₆	35MCUM
25-Dimethylcumene	C ₁₁ H ₁₆	25MCUM
24-Diethyltoluene	C ₁₁ H ₁₆	24ETOLU
35-Diethyltoluene	C ₁₁ H ₁₆	35ETOLU
Diphenyl	C ₁₂ H ₁₀	DPHNYL
Acenaphthene	C ₁₂ H ₁₀	ACNAPHT
12-Dimethylnaphthalene	C ₁₂ H ₁₂	12MNAPHT
13-Dimethylnaphthalene	C ₁₂ H ₁₂	13MNAPHT
14-Dimethylnaphthalene	C ₁₂ H ₁₂	14MNAPHT
15-Dimethylnaphthalene	C ₁₂ H ₁₂	15MNAPHT
16-Dimethylnaphthalene	C ₁₂ H ₁₂	16MNAPHT
17-Dimethylnaphthalene	C ₁₂ H ₁₂	17MNAPHT
23-Dimethylnaphthalene	C ₁₂ H ₁₂	23MNAPHT
26-Dimethylnaphthalene	C ₁₂ H ₁₂	26MNAPHT
27-Dimethylnaphthalene	C ₁₂ H ₁₂	27MNAPHT
1-Ethylnaphthalene	C ₁₂ H ₁₂	1ENAPHT
2-Ethylnaphthalene	C ₁₂ H ₁₂	2ENAPHT
2a345-Tetra-H-Acenaphten	C ₁₂ H ₁₄	TACNAPHT
Cyhexylbenzene	C ₁₂ H ₁₆	CYHXBZ

LONG NAME	FORMULA	SHORT NAME
12-Di-I-Propylbenzene	C12H18	12IPBZ
13-Di-I-Propylbenzene	C12H18	13IPBZ
14-Di-I-Propylbenzene	C12H18	14IPBZ
1-Phenylhexane	C12H18	1PHC6
C-Bicyhexyl	C12H22	CBCYHXYL
T-Bicyhexyl	C12H22	TBCYHXYL
Acenaphthylene	C12H8	ACNPHTYL
Fluorene	C13H10	FLUORENE
Diphenylmethane	C13H12	DPHC1
23dihyd-1h-Benz(E)Indene	C13H12	BZINDENE
3-Methyl-Biphenyl	C13H12	3MBPHNYL
1-N-Propylnaphthalene	C13H14	1NPNAPHT
2-N-Propylnaphthalene	C13H14	2NPNAPHT
Hexahydro-Furanene	C13H16	HHFURA
Anthracene	C14H10	ANTHRA
Phenanthrene	C14H10	PHANTH
35-Dimethyl-Biphenyl	C14H14	35MBPHNL
34-Dimethyl-Biphenyl	C14H14	34MBPHNL
33-Dimethylbiphenyl	C14H14	33DMBP
1-N-Butylnaphthalene	C14H16	1BNAPHT
2-N-Butylnaphthalene	C14H16	2BNAPHT
Tetradecahydroanthracene	C14H24	TDHANTHR
Tetradecahydrophenantren	C14H24	TDHPHNTR
1-Methylantracene	C15H12	1MANTHRA
Guaiazulene	C15H18	GUAIAZ
Pyrene	C16H10	PYRENE
Fluoranthene	C16H10	FLANTHEN
Chrysene	C18H12	CHRYSENE
Benz(A)Anthracene	C18H12	BZANTHRA
Naphthacene	C18H12	NPHTCN
Triphenylene	C18H12	TPHENE
3,4-Benzo(C)Phenanthrene	C18H12	34BPT
1,2,4-Triethylbenzene	C18H12	123TEB

LONG NAME	FORMULA	SHORT NAME
1,3,5triethylbenzene	C18H12	135TEB
O-Terphenyl	C18H14	OTPHNYL
M-Terphenyl	C18H14	MTPHNYL
P-Terphenyl	C18H14	PTPHNYL
1-C1-7-I-C3-Phenanthrene	C18H18	PHENANTH
Octadecahydrochrysene	C18H30	ODHCHRY
Benzo(A)-Pyrene	C20H12	BZPYRENE
7,12-Dimet.Benz.(A)Anthr	C20H16	712DMBA
9-Cy.Hexyltetrahyd.Anthr	C20H34	9CYHTHA
Coronene	C24H12	CORONENE

ESTERS

LONG NAME	FORMULA	SHORT NAME
Methyl Formate	C2H4O2	MC1AT
Ethyl Formate	C3H6O2	EC1AT
Methyl Acetate	C3H6O2	MC2AT
Vinyl Acetate	C4H6O2	VC2AT
Propylene Carbonate	C4H6O3	PCARBOAT
Dimethyl Oxalate	C4H6O4	DMOXAT
Methyl Acrylate	C4H7O2	MACRYLAT
Ethyl Acetate	C4H8O2	EC2AT
Methyl Propionate	C4H8O2	MC3AT
N-Propyl Formate	C4H8O2	N-PC1AT
Ethyl Acrylate	C5H8O2	EACRYLAT
I-Butyl Formate	C5H10O2	I-BC1AT
N-Propyl Acetate	C5H10O2	N-PC2AT
Ethyl Propionate	C5H10O2	EC3AT
Methyl N-Butyrate	C5H10O2	MN-C4AT
Methyl I-Butyrate	C5H10O2	MI-C4AT
N-Butyl Acetate	C6H12O2	N-BC2AT
I-Butyl Acetate	C6H12O2	I-BC2AT
Ethyl N-Butyrate	C6H12O2	EN-C4AT
Ethyl I-Butyrate	C6H12O2	EI-C4AT

LONG NAME	FORMULA	SHORT NAME
N-Propyl Propionate	C6H12O2	N-PC3AT
N-Butyl Acrylate	C7H12O2	BACRYLAT
Methyl Benzoate	C8H8O2	MBZAT
Ethyl Benzoate	C9H10O2	EBZAT
Butyl Benzoate	C11H14O2	BBZAT
Dibutyl-O-Phthalate	C16H22O4	DBPHTAT
Butyl Stearate	C22H44O2	BSTEARAT

ETHERS

LONG NAME	FORMULA	SHORT NAME
Ethylene Oxide	C2H4O1	C2OX
Dimethylether	C2H6O1	DME
Propylene Oxide	C3H6O1	C3OX
Vinylmethylether	C3H6O1	VME
Methylethylether	C3H8O1	MEE
Diethylether	C4H10O1	DEE
12-Dimethoxyethane	C4H10O2	12DMOE
Diethylene Glycol	C4H10O3	DEG
Furan	C4H4O1	FURAN
2,5-Dihydrofuran	C4H6O1	25DHF
Tetrahydrofuran	C4H8O1	THF
Vinylethylether	C4H8O1	VEE
12-Butylene Oxide	C4H8O1	12C4OX
Tetrahydropyran	C5H10O1	THP
Ethylpropylether	C5H12O1	EPE
Pyran	C5H6O1	PYRAN
Dihydropyran	C5H8O1	DHP
Ethylbutylether	C6H14O1	EBE
Diisopropylether	C6H14O1	DIPE
Dipropylether	C6H14O1	DPE
Methylphenylether	C7H8O1	MPE
Dibutylether	C8H18O1	DBE
Benzofuran	C8H6O1	BZFURAN

LONG NAME	FORMULA	SHORT NAME
Pyrogallol	C8H6O3	PYROGALL
2-Methylbenzofuran	C9H8O1	MBZFURAN
57-Dimethylbenzofuran	C10H10O1	57MBZF
Diphenylether	C12H10O1	DPHE
Dihexylether	C12H26O1	DHE
Naphthol(12.B)Furan	C12H6O1	NAPHFU
Dibenzofuran	C12H8O1	DBZFURAN

HALOGENATED DERIVATIVES

LONG NAME	FORMULA	SHORT NAME
1112-Tetrachloroethane	C2H2CL4	1112CLC2
111-Trichloroethane	C2H3CL3	111CLC2
111-Trifluoroethane	C2H3F3	C2F3
1122-Cl4-12-F2-Ethane	C2CL4F2	C2F2CL4
1122-Tetrachloroethane	C2H2CL4	1122CLC2
112-Trichloroethane	C2H3CL3	C2CL3
11-Cl2-1222-F4-Ethane	C2CL2F4	11C2F4CL
11-Dichloroethane	C2H4CL2	1C2CL2
11-Dichloroethylene	C2H2CL2	11CLC2-
11-Difluoroethane	C2H4F2	1C2F2
11-Difluoroethylene	C2H2F2	C2-F2
122-Cl3-112-F3-Ethane	C2CL3F3	C2F3CL3
123-Trichloropropane	C3H5CL3	C3CL3
12-Cl2-1122-F4-Ethane	C2CL2F4	12C2F4CL
12-Dichloroethane	C2H4CL2	2C2CL2
12-Dichloropropane	C3H6CL2	C3CL2
1-Chlorobutane	C4H9CL1	1C4CL
1-Cl-11-F2-Ethane	C2H3CL1F2	C2F2CL
2-Chlorobutane	C4H9CL1	2C4CL
Acetyl Chloride	C2H3O1CL1	C2OCL
Allyl Chloride	C3H5CL1	ALLCL
Boron Trichloride	CL3B1	CL3B
Boron Trifluoride	F3B1	F3B

LONG NAME	FORMULA	SHORT NAME
Bromobenzene	C6H5BR1	BRBZ
Bromochlorodifluorometha	C1BR1CL1F2	BRCLF2C1
Bromoethane	C2H5BR1	C2BR
Bromomethane	C1H3BR1	C1BR
Bromotrichloromethane	C1BR1CL3	BRCL3C1
C-12-Dichloroethylene	C2H2CL2	C12CLC2-
Carbon Tetrabromide	C1BR4	CBR4
Carbonic Bromide Fluorid	C1BR1F1O1	BRFCARB
Chlorobenzene	C6H5CL1	CLBZ
Chlorodifluoromethane	C1H1CL1F2	C1F2CL
Chloroethane	C2H5CL1	C2CL
Chloro-I-Propane	C3H7CL1	I-C3CL
Chloromethane	C1H3CL1	C1CL
Chloropentafluoroethane	C2CL1F5	C2CLF5
Chloroprene	C4H5CL1	CLPRENE
Chloropropane	C3H7CL1	C3CL
Chlorotrifluoroethylene	C2CL1F3	CF3CL
Chlorotrifluoromethane	C1CL1F3	C1F3CL
Cyanogen	C2N2	C2N2
Cyanogen Bromide	C1N1BR1	C1NBR
Cyanogen Chloride	C1N1CL1	C1NCL
Cyanogen Iodide	C1N1I1	C1NI
Dibromomethane	C1H2BR2	C1BR2
Dichlorodifluoromethane	C1CL2F2	C1F2CL2
Dichloromethane	C1H2CL2	C1CL2
Dichloromonofluoromethan	C1H1CL2F1	C1CL2F
Ethyl Iodide	C2H5I1	C2I
Fluorobenzene	C6H5F1	FBZ
Fluoroethane	C2H5F1	C2F
Fluoromethane	C1H3F1	C1F
Hexachloroethane	C2CL6	HCLC2
Hexachloropropylene	C3CL6	C3CL6
Hexafluorobenzene	C6F6	HXFBZ

LONG NAME	FORMULA	SHORT NAME
Hexafluoroethane	C2F6	C2F6
Hydrogen Bromide	H1BR1	HBR
Hydrogen Chloride	H1CL1	HCL
Hydrogen Fluoride	H1F1	HF
Hydrogen Iodide	H1I1	HI
Iodobenzene	C6H5I1	IBZ
Iodomethane	C1H3I1	C1I
M-Dichlorobenzene	C6H4CL2	M-DCLBZ
Nitrogen Trifluoride	N1F3	NF3
Nitrosyl Chloride	O1N1CL1	CLNO
Octafluoropropane	C3F8	C3F8
O-Dichlorobenzene	C6H4CL2	O-DCLBZ
P-Dichlorobenzene	C6H4CL2	P-DCLBZ
Pentachloroethane	C2H1CL5	PCLC2
Perfluoro-23-Dimethylc4	C6F14	PFDMC4
Perfluoro-2-Methylpentan	C6F14	PF2MC5
Perfluoro-3-Methylpentan	C6F14	PF3MC5
Perfluorocyhexane	C6F12	PFCYC6
Perfluoromethylcyc6	C7F14	PFMCYC6
Perfluoro-N-Heptane	C7F16	PFNC7
Perfluoro-N-Hexane	C6F14	PFNC6
Phosgene	C1O1CL2	COCL2
Phosphorus Trichloride	CL3P1	PCL3
Silicon Tetrachloride	CL4SI1	SICL4
Silicon Tetrafluoride	F4SI1	SIF4
Sulfur Hexafluoride	F6S1	SF6
T-12-Dichloroethylene	C2H2CL2	T12CLC2-
Tert-Butyl Chloride	C4H9CL1	TC4CL
Tetrachlorodiphenyl	C12H6CL4	TCLDPHYL
Tetrachloroethylene	C2CL4	C2CL4
Tetrachloromethane	C1CL4	CCL4
Tetrafluoroethene	C2F4	C2F4
Tetrafluoromethane	C1F4	CF4

LONG NAME	FORMULA	SHORT NAME
Trichloroethylene	C2H1CL3	C2-CL3
Trichlorofluoromethane	C1CL3F1	C1CL3F
Trichloromethane	C1H1CL3	C1CL3
Trifluorobromomethane	C1F3BR1	C1F3BR
Trifluoromethane	C1H1F3	CF3H
Vinyl Chloride	C2H3CL1	VCL
Vinyl Fluoride	C2H3F1	VF
Vinyl Formate	C3H4O2	VFORMAT

ISOCYANATE

LONG NAME	FORMULA	SHORT NAME
Methylisocyanate	C2H3O1N1	C1NCO

KETONES

LONG NAME	FORMULA	SHORT NAME
Ketene	C2H2O1	C2ONE
Acetone	C3H6O1	C3ONE
Gamma-Butyrolactone	C4H6O2	GBUTYONE
2-Butanone	C4H8O1	2C4ONE
Cypentanone	C5H8O1	C5ONE
2-Pentanone	C5H10O1	2C5ONE
3-Methyl-2-Butanone	C5H10O1	3MC4ONE
3-Pentanone	C5H10O1	3C5ONE
Cyhexanone	C6H10O1	CYC6ONE
4-Methyl-2-Pentanone	C6H12O1	42C5ONE
Methylphenylketone	C8H8O1	MPK

MISCELLANEOUS

LONG NAME	FORMULA	SHORT NAME
Argon	AR1	AR
Bromine	BR2	BR2

LONG NAME	FORMULA	SHORT NAME
Carbon	C1	C
Chlorine	CL2	CL2
Chlorine (Atomic)	CL1	CL
Deuterium	D2	D2
Fluorine	F2	F2
Helium 4	HE1	HE
Hydrogen	H2	H2
Iodine	I2	I2
Krypton	KR1	KR
Neon	NE1	NE
Nitrogen	N2	N2
Oxygen	O2	O2
Ozone	O3	O3
Radon	RN1	RN
Sulfur	S2	S2
Sulfur (S6)	S6	S6
Sulfur (S8)	S8	S8
Water	H2O1	H2O
Xenon	XE1	XE
13-Dioxopropadiene	C3O2	C3O2
14-Dioxane	C4H8O2	DIOXANE
Acetic Anhydride	C4H6O3	ACANH
Anisole	C7H8O1	ANISOLE
Ethylcellosolve	C4H10O2	ECSLVE
Maleic Anhydride	C4H2O3	MLANH
Oh Radical	H1O1	OH•
Phthalic Anhydride	C8H4O3	PHTANH

NAPHTENES

LONG NAME	FORMULA	SHORT NAME
Cypropane	C3H6	CYC3
Cybutane	C4H8	CYC4
Cypentane	C5H10	CYC5

LONG NAME	FORMULA	SHORT NAME
Cyhexane	C ₆ H ₁₂	CYC6
Methylcypentane	C ₆ H ₁₂	MCYC5
Cyheptane	C ₇ H ₁₄	CYC7
11-Dimethylcypentane	C ₇ H ₁₄	11CYC5
C-12-Dimethylcypentane	C ₇ H ₁₄	C12CYC5
T-12-Dimethylcypentane	C ₇ H ₁₄	T12CYC5
Ethylcypentane	C ₇ H ₁₄	ECYC5
Methylcyhexane	C ₇ H ₁₄	MCYC6
11-Dimethylcyhexane	C ₈ H ₁₆	11CYC6
C-12-Dimethylcyhexane	C ₈ H ₁₆	C12CYC6
T-12-Dimethylcyhexane	C ₈ H ₁₆	T12CYC6
C-13-Dimethylcyhexane	C ₈ H ₁₆	C13CYC6
T-13-Dimethylcyhexane	C ₈ H ₁₆	T13CYC6
C-14-Dimethylcyhexane	C ₈ H ₁₆	C14CYC6
T-14-Dimethylcyhexane	C ₈ H ₁₆	T14CYC6
Ethylcyhexane	C ₈ H ₁₆	ECYC6
112-Trimethylcypentane	C ₈ H ₁₆	112CC5
113-Trimethylcypentane	C ₈ H ₁₆	113CC5
C-C-T-124-Trimethylcyc5	C ₈ H ₁₆	CCTCC5
C-T-C-124-Trimethylcyc5	C ₈ H ₁₆	CTCCC5
1-Methyl-1-Ethylcypentan	C ₈ H ₁₆	11ECYC5
N-Propylcypentane	C ₈ H ₁₆	NPCYC5
I-Propylcypentan	C ₈ H ₁₆	IPCYC5
N-Propylcyhexane	C ₉ H ₁₈	NPCYC6
I-Propylcyhexane	C ₉ H ₁₈	IPCYC6
C-Decalin	C ₁₀ H ₁₈	C-DECA
T-Decalin	C ₁₀ H ₁₈	T-DECA
N-Butylcyhexane	C ₁₀ H ₂₀	NBCYC6
I-Butylcyhexane	C ₁₀ H ₂₀	IBCYC6
Sec-Butylcyhexane	C ₁₀ H ₂₀	SBCYC6
Ter-Butylcyhexane	C ₁₀ H ₂₀	TBCYC6
N-Hexylcypentane	C ₁₁ H ₂₂	NHXCC5
N-Heptycypentane	C ₁₂ H ₂₄	NHPCC5

LONG NAME	FORMULA	SHORT NAME
N-Octylcypentane	C13H26	NOCYC5
N-Nonylcypentane	C14H28	NNCYC5
N-Decylcypentane	C15H30	NDCYC5
N-Decylcyhexane	C16H32	NDCYC6
N-Dodecylcypentane	C17H34	NDDCYC5
N-Tridecylcypentane	C18H36	N3DCC5
N-Tetradecylcypentane	C19H38	N4DCC5
N-Pentadecylcypentane	C20H40	N5DCC5
N-Hexadecylcypentane	C21H42	N6DCC5

NITROGEN DERIVATIVES

LONG NAME	FORMULA	SHORT NAME
Ammonia	H3N1	NH3
Hydrazine	H4N2	N2H4
Hydrogen Cyanide	C1H1N1	HCN
Nitromethane	C1H3O2N1	C1NO2
Acetonitrile	C2H3N1	C2N
Ethylene Imine	C2H5N1	EIMINE
Nitroethane	C2H5O2N1	C2NO2
Acrylonitrile	C3H3N1	C3N
Acrolein	C3H4O1	ACROLEIN
Propionitrile	C3H5N1	PROPN
1-Nitropropane	C3H7O2N1	1C3NO2
2-Nitropropane	C3H7O2N1	2C3NO2
Pyrazine	C4H4N2	PYRAZINE
Allyl Cyanide	C4H5N1	ALLCN
Pyrrole	C4H5N1	PYRROLE
Butyronitrile	C4H7N1	BUTN
Pyrrolidine	C4H9N1	PYRRODIN
Piperidine	C5H11N1	PIPERIDI
2-Methyl-Pyrazine	C5H6N2	2MPYRAZN
N-Methyl-Pyrrole	C5H7N1	NMPYRR
Nitrobenzene	C6H5O2N1	BZNO2

LONG NAME	FORMULA	SHORT NAME
Phenylhydrozine	C ₆ H ₈ N ₂	PHHYDRZN
25-Dimethyl-Pyrrole	C ₆ H ₉ N ₁	25MPYRR
Benzonitrile	C ₇ H ₅ N ₁	BZN
O-Tolunitrile	C ₈ H ₇ N ₁	O-TOLN
Indole	C ₈ H ₇ N ₁	INDOLE
M-Cyanotoluene	C ₈ H ₇ N ₁	MCYATOLU
P-Cyanotoluene	C ₈ H ₇ N ₁	PCYATOLU
Quinoline	C ₉ H ₇ N ₁	QUINOL
I-Quinoline	C ₉ H ₇ N ₁	I-QUINOL
2-Methylindole	C ₉ H ₉ N ₁	2MINDOLE
1234-Tetrahydroquinoline	C ₉ H ₁₁ N ₁	1234THQ
Nn-Dimethyl-O-Toluidine	C ₉ H ₁₃ N ₁	NNDTOL
2-Methylquinoline	C ₁₀ H ₉ N ₁	2MQUINOL
4-Methylquinoline	C ₁₀ H ₉ N ₁	4MQUINOL
8-Methylquinoline	C ₁₀ H ₉ N ₁	8MQUINOL
Quinaldine	C ₁₀ H ₉ N ₁	QUINAL
Lepidine	C ₁₀ H ₉ N ₁	LEPIDINE
7-Methylquinoline	C ₁₀ H ₉ N ₁	7MQ
1-Methyl-I-Quinoline	C ₁₀ H ₉ N ₁	1M-I-Q
3-Methyl-I-Quinoline	C ₁₀ H ₉ N ₁	3M-I-Q
Caprylonitrile	C ₁₀ H ₁₉ N ₁	CAPN
24-Dimethylquinoline	C ₁₁ H ₁₁ N ₁	24DMQ
26-Dimethylquinoline	C ₁₁ H ₁₁ N ₁	26DMQ
Carbazole	C ₁₂ H ₉ N ₁	CARBAZOL
246-Trimethylquinoline	C ₁₂ H ₁₃ N ₁	246TEQ
Acridine	C ₁₃ H ₉ N ₁	ACRIDINE

OLEFINS

LONG NAME	FORMULA	SHORT NAME
Ethylene	C ₂ H ₄	C2-
Propadiene	C ₃ H ₄	C3--
Propylene	C ₃ H ₆	C3-
Vinylacetylene	C ₄ H ₄	VACTL

LONG NAME	FORMULA	SHORT NAME
12-Butadiene	C ₄ H ₆	12C ₄ --
13-Butadiene	C ₄ H ₆	13C ₄ --
1-Butene	C ₄ H ₈	1C ₄ -
C-2-Butene	C ₄ H ₈	C-2C ₄ -
T-2-Butene	C ₄ H ₈	T-2C ₄ -
I-Butylene	C ₄ H ₈	I-C ₄ -
13-Cypentadiene	C ₅ H ₆	13CYC ₅ -
Cypentene	C ₅ H ₈	CY-C ₅ -
12-Pentadiene	C ₅ H ₈	12C ₅ --
1-T-3-Pentadiene	C ₅ H ₈	1T3C ₅ --
14-Pentadiene	C ₅ H ₈	14C ₅ --
2-Methyl-13-Butadiene	C ₅ H ₈	2MC ₄ --
3-Methyl-12-Butadiene	C ₅ H ₈	3MC ₄ --
1-Pentene	C ₅ H ₁₀	1C ₅ -
C-2-Pentene	C ₅ H ₁₀	C-2C ₅ -
T-2-Pentene	C ₅ H ₁₀	T-2C ₅ -
2-Methyl-1-Butene	C ₅ H ₁₀	2M1C ₄ -
2-Methyl-2-Butene	C ₅ H ₁₀	2M2C ₄ -
3-Methyl-1-Butene	C ₅ H ₁₀	3M1C ₄ -
13-Cyhexadiene	C ₆ H ₈	13CYC ₆ -
1-Methyl-Cypentene	C ₆ H ₁₀	1MCC ₅ -
3-Methyl-Cypentene	C ₆ H ₁₀	3MCC ₅ -
15-Hexadiene	C ₆ H ₁₀	15C ₆ -
Cyhexene	C ₆ H ₁₀	CYC ₆ -
1-Hexene	C ₆ H ₁₂	1C ₆ -
C-2-Hexene	C ₆ H ₁₂	C-2C ₆ -
T-2-Hexene	C ₆ H ₁₂	T-2C ₆ -
C-3-Hexene	C ₆ H ₁₂	C-3C ₆ -
T-3-Hexene	C ₆ H ₁₂	T-3C ₆ -
2-Methyl-2-Pentene	C ₆ H ₁₂	2M2C ₅ -
3-Methyl-C-2-Pentene	C ₆ H ₁₂	3MC-2C ₅ -
3-Methyl-T-2-Pentene	C ₆ H ₁₂	3MT-2C ₅ -
4-Methyl-C-2-Pentene	C ₆ H ₁₂	4MC-2C ₅ -

LONG NAME	FORMULA	SHORT NAME
4-Methyl-T-2-Pentene	C ₆ H ₁₂	4MT-2C5-
23-Dimethyl-1-Butene	C ₆ H ₁₂	231C4-
23-Dimethyl-2-Butene	C ₆ H ₁₂	232C4-
33-Dimethyl-1-Butene	C ₆ H ₁₂	331C4-
1-Methyl-Cyhexene	C ₇ H ₁₂	1MCC6-
4-Methyl-Cyhexene	C ₇ H ₁₂	4MCC6-
1-Heptene	C ₇ H ₁₄	1C7-
233-Trimethyl-1-Butene	C ₇ H ₁₄	233C4-
14-Dimethylcyhexene	C ₈ H ₁₄	14MCC6-
1-Octene	C ₈ H ₁₆	1C8-
T-2-Octene	C ₈ H ₁₆	T2C8-
1-Nonene	C ₉ H ₁₈	1C9-
1-Decene	C ₁₀ H ₂₀	1C10-
1-Undecene	C ₁₁ H ₂₂	1C11-
1-Dodecene	C ₁₂ H ₂₄	1C12-
1-Tridecene	C ₁₃ H ₂₆	1C13-
1-Tetradecene	C ₁₄ H ₂₈	1C14-
1-Pentadecene	C ₁₅ H ₃₀	1C15-
1-Hexadecene	C ₁₆ H ₃₂	1HC10-
1-Octadecene	C ₁₈ H ₃₆	1C18-
1-Eicosene	C ₂₀ H ₄₀	C20-

OXIDES

LONG NAME	FORMULA	SHORT NAME
Carbon Dioxide	C1O2	CO2
Carbon Monoxide	C1O1	CO
Deuterium Oxide	O1D2	D2O
Dinitrogen Trioxide	O3N2	N2O3
Hydrogen Peroxide	H2O2	H2O2
Nitric Oxide	O1N1	NO
Nitrogen Dioxide	O2N1	NO2
Nitrogen Dioxide (N2o4)	O4N2	N2O4
Nitrogen Oxide N2o2	O2N2	N2O2

LONG NAME	FORMULA	SHORT NAME
Nitrogen Pentoxide	O5N2	N2O5
Nitrogen Trioxide	O3N1	NO3
Nitrous Oxide	O1N2	N2O
Sulfur Dioxide	O2S1	SO2
Sulfur Trioxide	O3S1	SO3

PARAFFINS

LONG NAME	FORMULA	SHORT NAME
Methane	C1H4	C1
Ethane	C2H6	C2
Propane	C3H8	C3
N-Butane	C4H10	N-C4
I-Butane	C4H10	I-C4
N-Pentane	C5H12	N-C5
2-Methylbutane	C5H12	2MC4
22-Dimethylpropane	C5H12	22DMC3
N-Hexane	C6H14	N-C6
2-Methylpentane	C6H14	2MC5
3-Methylpentane	C6H14	3MC5
22-Dimethylbutane	C6H14	22DMC4
23-Dimethylbutane	C6H14	23DMC4
N-Heptane	C7H16	N-C7
2-Methylhexane	C7H16	2MC6
3-Methylhexane	C7H16	3MC6
22-Dimethylpentane	C7H16	22DMC5
23-Dimethylpentane	C7H16	23DMC5
24-Dimethylpentane	C7H16	24DMC5
33-Dimethylpentane	C7H16	33DMC5
3-Ethylpentane	C7H16	3EC5
223-Trimethylbutane	C7H16	223MC4
N-Octane	C8H18	N-C8
2-Methylheptane	C8H18	2MC7
3-Methylheptane	C8H18	3MC7

LONG NAME	FORMULA	SHORT NAME
4-Methylheptane	C ₈ H ₁₈	4MC7
22-Dimethylhexane	C ₈ H ₁₈	22DMC6
23-Dimethylhexane	C ₈ H ₁₈	23DMC6
24-Dimethylhexane	C ₈ H ₁₈	24DMC6
25-Dimethylhexane	C ₈ H ₁₈	25DMC6
33-Dimethylhexane	C ₈ H ₁₈	33DMC6
34-Dimethylhexane	C ₈ H ₁₈	34DMC6
3-Ethylhexane	C ₈ H ₁₈	3EC6
223-Trimethylpentane	C ₈ H ₁₈	223MC5
224-Trimethylpentane	C ₈ H ₁₈	224MC5
233-Trimethylpentane	C ₈ H ₁₈	233MC5
234-Trimethylpentane	C ₈ H ₁₈	234MC5
2-Methyl-3-Ethylpentane	C ₈ H ₁₈	2M3EC5
3-Methyl-3-Ethylpentane	C ₈ H ₁₈	3M3EC5
N-Nonane	C ₉ H ₂₀	N-C9
223-Trimethylhexane	C ₉ H ₂₀	223MC6
224-Trimethylhexane	C ₉ H ₂₀	224MC6
225-Trimethylhexane	C ₉ H ₂₀	225MC6
33-Diethylpentane	C ₉ H ₂₀	33DEC5
2233-Tetramethylpentane	C ₉ H ₂₀	2233C5
2234-Tetramethylpentane	C ₉ H ₂₀	2234C5
2244-Tetramethylpentane	C ₉ H ₂₀	2244C5
2334-Tetramethylpentane	C ₉ H ₂₀	2334C5
N-Decane	C ₁₀ H ₂₂	N-C10
335-Trimethylheptane	C ₁₀ H ₂₂	335MC7
2233-Tetramethylhexane	C ₁₀ H ₂₂	2233C6
2255-Tetramethylhexane	C ₁₀ H ₂₂	2255C6
N-Undecane	C ₁₁ H ₂₄	N-C11
N-Dodecane	C ₁₂ H ₂₆	N-C12
N-Tridecane	C ₁₃ H ₂₈	N-C13
N-Tetradecane	C ₁₄ H ₃₀	N-C14
N-Pentadecane	C ₁₅ H ₃₂	N-C15
N-Hexadecane	C ₁₆ H ₃₄	N-C16

LONG NAME	FORMULA	SHORT NAME
N-Heptadecane	C17H36	N-C17
N-Octadecane	C18H38	N-C18
N-Nonadecane	C19H40	N-C19
N-Eicosane	C20H42	N-C20
2,3-Dimethyloctadecane	C20H42	23MC18
2,4-Dimethyloctadecane	C20H42	24MC18
2-Methylnonadecane	C20H42	2MC19
3-Methylnonadecane	C20H42	3MC20
4-Methylnonadecane	C20H42	4MC21
5-Methylnonadecane	C20H42	5MC22
2,4,6-Trimethylheptadeca	C20H42	246C17
2,3-Dimethylnonadecane	C21H44	23MC19
2,4-Dimethylnonadecane	C21H44	24MC19
2-Methyleicosane	C21H44	2MC20
3-Methyleicosane	C21H44	3MC21
4-Methyleicosane	C21H44	4MC22
5-Methyleicosane	C21H44	5MC23
2,4,6-Trimethyloctadecan	C21H44	246C18
N-Uneicosane	C21H44	N-C21
Docosane	C22H46	N-C22
Tricosane	C23H48	N-C23
Tetracosane	C24H50	N-C24
Pentacosane	C25H52	N-C25
Hexacosane	C26H54	N-C26
Heptacosane	C27H56	N-C27
Octacosane	C28H58	N-C28
Nonacosane	C29H60	N-C29
triacontane	C30H62	N-C30
Untriacontane	C31H64	N-C31
Dotriacontane	C32H66	N-C32
Tritriacontane	C33H68	N-C33
Tetratriacontane	C34H70	N-C34
Pentatriacontane	C35H72	N-C35

LONG NAME	FORMULA	SHORT NAME
Hexatriacontane	C36H74	N-C36
Heptatriacontane	C37H76	N-C37
Octatriacontane	C38H78	N-C38
Nonatriacontane	C39H80	N-C39
Tetracontane	C40H82	N-C40
Pentacontane	C50H102	N-C50
Hexacontane	C60H122	N-C60
Heptacontane	C70H142	N-C70
Octacontane	C80H162	N-C80
Nonacontane	C90H182	N-C90
Hectane	C100H202	C100

SULFUR DERIVATIVES

LONG NAME	FORMULA	SHORT NAME
Hydrogen Sulfide	H2S1	H2S
Carbonyl Sulfide	C1O1S1	COS
Carbon Disulfide	C1S2	CS2
Methanethiol	C1H4S1	C1SH
Ethanethiol	C2H6S1	C2SH
Dimethylsulfide	C2H6S1	DMS
Methylethylsulfide	C3H8S1	MES
Ethylene Sulfide	C2H4S1	C2S
Dimethyl Disulfide	C2H6S2	C2S2
Thia-Cybutane	C3H6S1	THCYC4
Thiophene	C4H4S1	THIOPH
11-Diox-Tetrahydrothioph	C4H8O2S1	11DTHTPH
3-Methyl-Thiacybutane	C4H8S1	3MTB
Tetrahydrothiophene	C4H8S1	THTHIOPH
Diethylsulfide	C4H10S1	DES
Diethyldisulfide	C4H10S2	DEDS
2-Methylthiophene	C5H6S1	2MTIOPH
3-Methylthiophene	C5H6S1	3MTIOPH
Tetrohydro-Thiopyran	C5H10S1	THTPRYAN

LONG NAME	FORMULA	SHORT NAME
2-Methyl-Thia-Cypentane	C ₅ H ₁₀ S ₁	2MTCYC5
3-Methyl-Thia-Cypentane	C ₅ H ₁₀ S ₁	3MTCYC5
Cypentanethiol	C ₅ H ₁₀ S ₁	CYC5TH
Benzenethiol	C ₆ H ₆ S ₁	BZTHIOL
3-Ethyl-Thiophene	C ₆ H ₈ S ₁	3ETHIOPH
23-Dimethyl-Thiophene	C ₆ H ₈ S ₁	23MTHIOP
24-Dimethyl-Thiophene	C ₆ H ₈ S ₁	24MTHIOP
25-Dimethyl-Thiophene	C ₆ H ₈ S ₁	25MTHIOP
34-Dimethyl-Thiophene	C ₆ H ₈ S ₁	34MTHIOP
1-Cypenthy-1-Thiaethane	C ₆ H ₁₂ S ₁	1CYP1TT
3-Methyl-Thia-Cyhexane	C ₆ H ₁₂ S ₁	3MTCYC6
2-Methyl-Thia-Cyhexane	C ₆ H ₁₂ S ₁	2MTCYC6
4-Methyl-Thia-Cyhexane	C ₆ H ₁₂ S ₁	4MTCYC6
Cyhexane-Thiol	C ₆ H ₁₂ S ₁	CYC6THOL
1-Phenyl-1-Thiaethane	C ₇ H ₈ S ₁	1P1TT
2-Methyl-Benzenethiol	C ₇ H ₈ S ₁	2MBTHIOL
4-Ethyl-2-Methylthiophen	C ₇ H ₁₀ S ₁	42THIOPN
5-Ethyl-2-Methylthyophen	C ₇ H ₁₀ S ₁	52THIOPN
Benzothiophene	C ₈ H ₆ S ₁	BZTHIOP
1-Thiaindan	C ₈ H ₈ S ₁	1THIAIDN
23-Dihydro-Benzothiophen	C ₈ H ₈ S ₁	23DHBTH
Benzylmethylsulfide	C ₈ H ₁₀ S ₁	BMS
2-Thiatetralin	C ₉ H ₁₀ S ₁	2THTTLIN
2-Ethyl-1-Thiaindene	C ₁₀ H ₁₀ S ₁	2E1THIAI
12-Dicypentylsulfide	C ₁₀ H ₁₈ S ₂	DCYPS2
Dibenzothiophene	C ₁₂ H ₈ S ₁	DBZTHIOP
Phenylsulfide	C ₁₂ H ₁₀ S ₁	PHSULFD

SECTION 8: SOLIDS DATA BASE

The properties stored for each compound are listed in the following table.

The characteristics of the solids are defined with regards to a quantity of solid ("unit") specified by the user and of which the weight is given by the parameter UNITW (kg/unit). In fact, UNITW corresponds to the molecular weight for chemical compounds that are perfectly determined.

PARAMETER	DESCRIPTION	PHYSICAL UNITS
H0FS	Standard enthalpy of formation	kJ/unit
G0FS	Gibbs standard energy of formation	kJ/unit
UNITW	Mass of selected solid unit	kg/unit
CRIST	Cristal Type 1 : triclinic 2 : monoclinic 3 : orthorhombic 4 : rhombohedral 5 : quadratic 6 : hexagonal 7 : simple cubic 8 : face centered cubic 9 : cubic with $T > T(\text{Curie})$ 10: centered cubic 11: centered cubic with $T > T(\text{Curie})$	-
TMIN(PS) TMAX(PS)	Temperature range for sublimation pressure	K
PSA, PSB, PSC, PSD	Sublimation pressure $\ln(\text{PS}) = A + B/(T + C) + D \cdot T$	bar
TMIN(CS) TMAX(CS)	Temperature range for heat capacity	K
CSA, CSB CSC, CSD	Heat capacity $\text{CS} = A + B \cdot T + C \cdot T^2 + D/T^2$	kJ/K/unit

PARAMETER	DESCRIPTION	PHYSICAL UNITS
TMIN(VS) TMAX(VS)	Temperature range for volume	K
VSA, VSB, VSC	Unit volume $VS = A + B \cdot T + C \cdot T^2$	m ³ /unit
TMIN(CT) TMAX(CT)	Temperature range for heat conductivity	K
CONDS1, CONDS2, CONDS3	Heat conductivity $CONDS = S1 + S2 \cdot T + S3 \cdot T^2$	KW/m/K

The list of compounds available in this library is shown below.

LONG NAME	FORMULA	SHORT NAME
Aluminum	AL1	AL
Copper	CU1	CU
Gold1	AU1	AU_1
Alpha1 Iron	FE1	A_FE
Silver1	AG1	AG_1
Manganese Hydroxyoxyd	MN1O2H1	MNO2H
Alpha-Cobalt	CO1	A_CO
Beta1 Cobalt	CO1	B_CO
Gamma1 Cobalt	CO1	G_CO
Cobalt Silicide	CO1SI1	COSI
Beta Cobalt Oxide	CO1O1	B_(CO)O
Alpha Cobalt Oxide	CO1O1	A_(CO)O
Copper(I) Chloride	CU1CL1	CUCL
Copper(II) Chloride	CU1CL2	CUCL2
Copper(II) Oxide	CU1O1	CUO
Copper(I) Oxide	CU2O1	CU2O
Nickel1	NI1	NI_1
Nickel Sulfate	NI1S1O4	NISO4
Alpha Nickel Monosulfide	NI1S1	NIS
Silver Fluoride	AG1F1	AGF

LONG NAME	FORMULA	SHORT NAME
Silver Chloride	AG1CL1	AGCL
Silver Bromide	AG1BR1	AGBR
Alpha Silver Iodide	AG1I1	A_AGI
Beta Silver Iodide	AG1I1	B_AGI
Silver Oxide	AG2O1	AG2O
Alpha Silver Sulfide	AG2S1	A_AG2S
Beta Silver Sulfide	AG2S1	B_AG2S
Gamma Silver Sulfide	AG2S1	G_AG2S
Alpha Silver Sulfate	AG2S1O4	A_2AGSO4
Beta Silver Sulfate	AG2S1O4	B_2AGSO4
Alpha Silver Selenide	AG2SE1	A_SEAG2
Beta Silver Selenide	AG2SE1	B_SEAG2
Alpha Silver Telluride	AG2TE1	A_TEAG2
Beta Silver Telluride	AG2TE1	B_TEAG2
Trisilver Antimonide	AG3SB1	AG3SB
Silver Carbonate	AG2C1O3	AG2CO3
Alpha Silver Nitrate	AG1N1O3	A_AGNO3
Beta Silver Nitrate	AG1N1O3	B_AGNO3
Silver Tungstate	AG2O1.W1O3	WO3.AG2O
Alpha1 Aluminum Fluoride	AL1F3	A_ALF3
Beta Aluminum Fluoride	AL1F3	B_ALF3
Aluminum Chloride	AL1CL3	ALCL3
Aluminum Bromide	AL1BR3	ALBR3
Aluminum Iodide	ALI3	ALI3
Alpha1 Aluminum Oxide	AL2O3	A_AL2O3
Gamma Aluminum Oxide	AL2O3	G_AL2O3
Al Oxide Monohydrate	AL2O3.H2O1	H2O.AL2O
Al Oxide Trihydrate	AL2O3.H6O3	H6O3.AL2
Aluminum Hydroxide	AL1O3H3	AL(OH)3
Aluminum Sulfate	AL2S3O12	AL2S3O12
Aluminum Nitride1	AL1N1	ALN
Tetraaluminum Tricarbide	AL4C3	AL4C3
Aluminum Disilicate	AL2O3.SI2O4	SI2.AL2O

LONG NAME	FORMULA	SHORT NAME
Al Disilicate Dihydrate	AL2O3.SI2O4.H4O2	AL.SI2.H
Al Silicate: Andalusite	AL2O3.SI1O2	ANDLSITE
Al Silicate: Kyanite	AL2O3.SI1O2	KYANITE
3-Al 2-Si: Mullite2	AL6O9.SI2O4	MULLITE2
Al Silicate: Sillimanite	AL2O3.SI1O2	SILLIMAN
2-Al Titanium 5-Oxide	AL2O3.TI1O2	TI.AL2O3
As Trioxide Claudetite	AS2O3	CLAUDETI
As Trioxide Arsenolite	AS2O3	ARSENOLI
Arsenic Pentoxide	AS2O5	AS2O5
Arsenic	AS1	AS
1-Ca 1-S 2-H2o: Gypsum	CA1S1O4.H4O2	GYPSUM
Gold Trifluoride	AU1F3	AUF3
Gold Monochloride	AU1CL1	AUCL
Gold Trichloride	AU1CL3	AUCL3
Gold Bromide	AU1BR1	AUBR
Gold Monoiodide	AU1I1	AUI
Gold Diantimonide	AU1SB2	AUSB2
Gold Monostannide	AU1SN1	AUSN
Boron1	B1	B_1
Boron: Amorphous	B1	B(AMRPH)
Boron Oxide	B2O3	B2O3
Alpha Boron Nitride	B1N1	A_BN
Tetraboron Monocarbide	B4C1	B4C
Boron2	B1	B_2
Beta Boron Nitride	B1N1	B_BN
Alpha Barium	BA1	A_BA
Barium	BA1	BA
Barium Fluoride	BA1F2	BAF2
Alpha Barium Chloride	BA1CL2	A_BACL2
Beta Barium Chloride	BA1CL2	B_BACL2
Barium Bromide	BA1BR2	BABR2
Barium Iodide	BA1I2	BAI2
Barium Oxide	BA1O1	BAO

LONG NAME	FORMULA	SHORT NAME
Barium Sulfate	BA1S1O4	BASO4
Alpha Barium Carbonate	BA1C1O3	A_BACO3
Beta Barium Carbonate	BA1C1O3	B_BACO3
Gamma Barium Carbonate	BA1C1O3	G_BACO3
Silver Bromate	AG1BR1O3	AGBRO3
Silver 1.64-Telluride1	AG1.64TE1	AG.TE1
Silver 1.64-Telluride2	AG1.64TE1	AG.TE2
Silver 1.64-Telluride3	AG1.64TE1	AG.TE3
Silver 1.64-Telluride4	AG1.64TE1	AG.TE4
Silver 1.64-Telluride5	AG1.64TE1	AG.TE5
Silver 1.64-Telluride6	AG1.64TE1	AG.TE6
Silver Diphosphide	AG1P2	AGP2
Silver Triphosphide	AG1P3	AGP3
Silver Chromate	AG2CR1O4	AG2CRO4
Aluminum Hydride	AL1H3	ALH3
Aluminum Oxychloride	AL1O1CL1	ALOCL
Al Chloride Hexahydrate	AL1CL3.H12O6	H.ALCL3
Aluminum Sulfide	AL2S3	AL2S3
Aluminum Selenide	AL2SE3	AL2SE3
Aluminum Telluride	AL2TE3	AL2TE3
Aluminum Monophosphide	AL1P1	ALP
Alpha Aluminum Phosphate	AL1P1O4	A_ALPO4
Beta Aluminum Phosphate	AL1P1O4	B_ALPO4
Gamma Aluminum Phosphate	AL1P1O4	G_ALPO4
Aluminum Arsenide	AL1AS1	ALAS
Aluminum Antimonide	AL1SB1	ALSB
Kaolinite	AL2O3.SI2O4.H4O2	KAOLNITE
Aluminum Diboride	AL1B2	ALB2
Aluminum 12-Boride	AL1B12	AL12-B
18-Al 4-Boron 33-Oxide	AL18O27.B4O6	AL18.B4
4-Al 2-Boron 9-Oxide	AL4O6.B2O3	AL4.B2O3
Arsenic Iodide	AS1I3	ASI3
Arsenic Monosulfide	AS2S2	AS2S2

LONG NAME	FORMULA	SHORT NAME
Arsenic Sulfide	AS2S3	AS2S3
Arsenic Selenide	AS2SE3	AS2SE3
Arsenic Telluride	AS2TE3	AS2TE3
Gold Oxide	AU2O3	AU2O3
Alpha Au Monoselenide	AU1SE1	A_AUSE
Beta Au Monoselenide	AU1SE1	B_AUSE
Gold Ditelluride	AU1TE2	AUTE2
2-Au Triphosphide	AU2P3	AU2P3
Gold 2-Lead	AU1PB2	AUPB2
Boron Sulfide	B2S3	B2S3
Boron Monophosphide	B1P1	BP
Barium Peroxide	BA1O2	BAO2
Barium Hydroxide	BA1O2H2	BA(OH)2
Barium Sulfide	BA1S1	BAS
Barium Telluride	BA1TE1	BATE
Barium Nitride	BA3N2	BA3N2
Barium Nitrate	BA1N2O6	BA(NO3)2
2-Barium Tin	BA2SN1	BA2SN
Barium 3-Tin	BA1SN3	BASN3
2-Barium Lead	BA2PB1	BA2PB
Barium 3-Lead	BA1PB3	BAPB3
Barium Tungstate	BA1O1.W1O3	WO3.BAO
Barium Uranate	BA1O1.U1O3	UO3.BAO
Barium Metazirconate	BA1O1.ZR1O2	ZRO2.BAO
Barium Hafnate	BA1O1.HF1O2	HFO2.BAO
Barium Meta Silicate	BA1O1.SI1O2	SIO2.BAO
Barium Orthosilicate	BA2O2.SI1O2	SI.BA2O2
Ba 2-Aluminum 4-Oxide	BA1O1.AL2O3	BAO.AL2
3-Ba 2-Aluminum 6-Oxide	BA3O3.AL2O3	AL2.BA3O
Alpha Ba Ti 3-Oxide	BA1O1.TI1O2	A_TLBAO
Beta Ba Ti 3-Oxide	BA1O1.TI1O2	B_TLBAO
2-Ba Ti 4-Oxide	BA2O2.TI1O2	TI.BA2O2
Ba Molybdenum 4-Oxide	BA1O1.MO1O3	MO.BAO

LONG NAME	FORMULA	SHORT NAME
Beryllium1	BE1	BE_1
Alpha Beryllium Fluoride	BE1F2	A_BE1F2
Beta Beryllium Fluoride	BE1F2	B_BE1F2
Beryllium Chloride	BE1CL2	BECL2
Beryllium Bromide	BE1BR2	BEBR2
Beryllium Iodide	BE1I2	BEI2
Alpha1 Be Oxide	BE1O1	A1_BEO
Alpha2 Be Oxide	BE1O1	A2_BEO
Beta Beryllium Oxide	BE1O1	B_BEO
Alpha Be Hydroxide	BE1O2H2	A_BE1O2H2
Beta Be Hydroxide	BE1O2H2	B_BE1O2H2
Alpha Be Sulfate	BE1S1O4	A_BE1S1O4
Beta Be Sulfate	BE1S1O4	B_BE1S1O4
Gamma Be Sulfate	BE1S1O4	G_BE1S1O4
Alpha Be Nitride	BE3N2	A_BE3N2
Beryllium Carbide	BE2C1	BEC
Be Ortho-Si : Phenacite	BE2O2.Si1O2	BE2.Si1O2
Triberyllium Diborate	BE3O3.B2O3	BE3.B2O3
Be 2-Aluminum 4-Oxide	BE1O1.AL2O3	AL2.BEO
Be 6-Aluminum 10-Oxide	BE1O1.AL6O9	AL6.BEO
Beryllium Tungstate	BE1O1.W1O3	WO3.BEO
Beryllium Monosulfide	BE1S1	BES
Be Sulfate Dihydrate	BE1S1O4.H4O2	BESO4.H4
Bismuth	BI1	BI
Bismuth Fluoride	BI1F3	BIF3
Bismuth Chloride	BI1CL3	BICL3
Bismuth Bromide	BI1BR3	BIBR3
Alpha Bi Monoiodide	BI1I1	A_BI1I1
Beta Bi Monoiodide	BI1I1	B_BI1I1
Bismuth Sulfate	BI2S3O12	BI2SO43
Bismuth Telluride	BI1TE3	TE3BI
Alpha Bismuth Oxide	BI2O3	A_BI2O3
Beta Bismuth Oxide	BI2O3	B_BI2O3

LONG NAME	FORMULA	SHORT NAME
Bismuth Sulfide	BI2S3	BI2S3
Bismuth Selenide	BI2SE3	BI2SE3
Alpha2 Aluminum Oxide	AL2O3	A2_AL2O3
3-Al 2-Si: Mullite2	AL6O9.SI2O4	MULLITE2
Aluminum Nitride2	AL1N1	2ALN
Aluminum Nitride3	AL1N1	3ALN
Silver2	AG1	AG_2
Alpha2 Aluminum Fluoride	AL1F3	ALF3
Gold2	AU1	AU_2
Boron3	B1	B_3
Beryllium2	BE1	BE_2
Alpha Calcium	CA1	A_CA
Beta Calcium	CA1	B_CA
Gamma Calcium	CA1	G_CA
Calcium Hydride	CA1H2	CAH2
Alpha Calcium Fluoride	CA1F2	A_CAF2
Beta Calcium Fluoride	CA1F2	B_CAF2
Calcium Chloride	CA1CL2	CACL2
Calcium Bromide	CA1BR2	CABR2
Calcium Iodide	CA1I2	CAI2
Calcium Oxydichloride	CA1O1CL2	CAOCL2
Calcium Oxyde	CA1O1	CAO
Calcium Peroxyde	CA1O2	CAO2
Calcium Hydroxide	CA1O2H2	CA(OH)2
Calcium Sulfide	CA1S1	CAS
Calcium Sulfite	CA1S1O3	CASO3
Calcium Sulfate	CA1S1O4	CASO4
Alpha 1-Ca 1-S 1/2-Hydra	CA1S1O4.H1O0.5	CA.HYDRA
Calcium Selenide	CA1SE1	CASE
Calcium Telluride	CA1TE1	CATE
Tricalcium Dinitride	CA3N2	CA3N2
Calcium Nitrate	CA1N2O6	CAN2O6
Tricalcium Diphosphide	CA3P2	CA3P2

LONG NAME	FORMULA	SHORT NAME
Alpha Ca Orthophosphate	CA3P2O8	A_CA3P2
Beta Ca Orthophosphate	CA3P2O8	B_CA3P2
Alpha Ca Pyrophosphate	CA2P2O7	A_CA2P2
Beta Ca Pyrophosphate	CA2P2O7	B_CA2P2
Cadmium	CD1	CD_1
Cadmium Sulfide	CD1S1	CDS
Cadmium Carbonate	CDC1O3	CDCO3
Cadmium Hydroxyde	CD1O2H2	CD(OH)2
Carbon: Graphite	C1	C_1
Carbon: Graphite	C1	C_2
Carbon: Diamond	C1	C_3
Beta2 Cobalt	CO1	B2_CO
Gamma2 Cobalt	CO1	G2_CO
Cobalt Dichloride	CO1CL2	COCL2
Cobalt(2,3) Oxide	CO3O4	CO3O4
Cobalt Hydroxyde	CO1O2H2	CO(OH)2
Cobalt(Ii) Sulfate	CO1S1O4	COSO4
Cobalt Carbonate	CO1C1O3	COCO3
Cobalt 0.89-Sulfide	CO1S0.89	COS0.89
Cobalt 1.333-Sulfide	CO1S1.333	COS1.333
Copper Hydroxide	CU1O2H2	CU(OH)2
Copper Carbonate	CU1C1O3	CUCO3
Copper(Ii) Sulfide	CU1S1	CUS
Iron(Ii) Carbonate	FE1C1O3	FECO3
Alpha2 Iron	FE1	A2_FE
Alpha3 Iron	FE1	A3_FE
Alpha4 Iron	FE1	A4_FE
Alpha5 Iron	FE1	A5_FE
Gamma Iron	FE1	G_FE
Delta Iron	FE1	D_FE
Fe(Iii) Oxide: Hematite	FE2O3	FE2O3
Beta Iron(Iii) Oxide	FE2O3	B_FE2O3
Gamma Iron(Iii) Oxide	FE2O3	G_FE2O3

LONG NAME	FORMULA	SHORT NAME
Iron(Ii) Hydroxide	FE1O2H2	FE(OH)2
Iron(Iii) Hydroxide	FE1O3H3	FEO3H3
Alpha Iron Monosulfide	FE1S1	A_FES
Beta Iron Monosulfide	FE1S1	B_FES
Gamma Iron Monosulfide	FE1S1	G_FES
Magnesium	MG1	MG
Magnesium Oxide	MG1O1	MGO
Magnesium Hydroxide	MG1O2H2	MG(OH)2
Magnesium Carbonate	MG1C1O3	MGCO3
Alpha1 Manganese	MN1	A1_MN
Alpha2 Manganese	MN1	A2_MN
Beta Manganese	MN1	B_MN
Gamma Manganese	MN1	G_MN
Delta Manganese	MN1	D_MN
Manganese(Ii) Oxide	MN1O1	MNO
Alpha Mn(Iv) Oxide	MN1O2	A_MNO2
Manganese Monosulfide	MN1S1	MNS
Manganese(Ii) Carbonate	MN1C1O3	MNCO3
Sodium	NA1	NA
Alpha Sodium Hydroxide	NA1O1H1	A_NAOH
Beta Sodium Hydroxide	NA1O1H1	B_NAOH
Disodium Monosulfide	NA2S1	NA2S
Alpha Sodium Carbonate	NA2C1O3	NA2CO3
Beta Sodium Carbonate	NA2C1O3	NA2CO3
Nickel2	NI1	NI_2
Nickel3	NI1	NI_3
Nickel4	NI1	NI_4
Nickel5	NI1	NI_5
Nickel6	NI1	NI_6
Beta Nickel Monosulfide	NI1S1	NIS
Nickel Carbonate	NI1C1O3	NICO3
Lead	PB1	PB
Lead Sulfide	PB1S1	PBS

LONG NAME	FORMULA	SHORT NAME
Alpha Sulfur	S1	A_S
Beta1 Sulfur	S1	B1_S
Beta2 Sulfur	S1	B2_S
Silicon	SI1	SI
Alpha Si Dioxide: Quartz	SI1O2	A_QUARTZ
Beta Si Dioxide: Quartz	SI1O2	B_QUARTZ
Alpha Si 2-O :Cristobali	SI1O2	A_CRISTO
Beta Si 2-O: Cristobalit	SI1O2	B_CRISTO
Vanadium1	V1	V_1
Vanadium2	V1	V_2
Vanadium3	V1	V_3
Zinc	ZN1	ZN
Zinc Oxide	ZN1O1	ZNO
Zinc Sulfide: Sphalerite	ZN1S1	ZNS(SPH)
Zinc Sulfide: Wurzite	ZN1S1	ZNS(WRZ)
Zinc Carbonate	ZN1C1O3	ZNCO3
Alpha Zirconium	ZR1	A_ZR
Beta Zirconium	ZR1	B_ZR
Ca Monohydrogen Phosphat	CA1H1P1O4	CAHPO4
Calcium Antimonide	CA3SB2	CA3SB2
Calcium Bismuthide	CA3BI2	CA3BI2
Alpha Calcium Dicarbide	CA1C2	A_CAC
Beta Calcium Dicarbide	CA1C2	B_CAC
Ca Carbonate: Aragonite	CA1C1O3	ARGNITE
Ca Carbonate: Calcite	CA1C1O3	CALCITE
Ca Mg Carbonate:Dolomite	CA1C1O3.MG1C1O3	DOLOMITE
2-Calcium Silicon	CA2SI1	2-CASI
Calcium Silicon	CA1SI1	CASI
Calcium 2-Silicon	CA1SI2	CA2-SI
Alpha Ca Meta-Si: Wollas	CA1O1.SI1O2	A_WOLLAS
Beta Ca Meta-Si: Wollast	CA1O1.SI1O2	B_WOLLAS
Ca Metasilicate:Pseudowo	CA1O1.SI1O2	PSEU_WOL
Gamma Ca Orthosilicate	CA2O2.SI1O2	G_ORTHSI

LONG NAME	FORMULA	SHORT NAME
Beta Ca Orthosilicate	CA2O2.SI1O2	B_ORTHSI
Epsilon Ca Orthosilicat	CA2O2.SI1O2	E_ORTHSI
Alpha Ca Orthosilicate	CA2O2.SI1O2	A_ORTHSI
Tricalcium Silicate	CA3O3.SI1O2	CA3.SI
3-Ca 2-Si: Frankinite	CA3O3.SI2O4	CA3.2SI
Ca Orthosi 7/6-Hydrate	CA2O2.SI1O2.H2.33O10.166	CA2.H2SI
3-Ca 2-Si 3-Hydrate	CA3O3.SI2O4.H6O3	CA3.H6SI
4-Ca 3-Si 3/2-Hydrate	CA4O4.SI3O6.H3O1.5	CA4.H3SI
6-Ca 6-Si Hydrate	CA6O6.SI6O12.H2O	CA6.H2SI
5-Ca 6-Si 3-Hydrate	CA5O5.SI6O12.H6O3	CA5.H6SI
5-Ca 6-Si 11/2-Hydrate	CA5O5.SI6O12.H11O5.5	CA5.H11
5-Ca 6-Si 21/2-Hydrate	CA5O5.SI6O12.H21O100.5	CA5.H21
Ca 2-Si 2-Hydrate	CA1O1.SI2O4.H4O2	CA1.H4SI
2-Ca 3-Si 5/2-Hydrate	CA2O2.SI3O6.H5O2.5	CA2.H5SI
Alpha 1-Ca 1-Ti 3-Oxide	CA1O1.TI1O2	A_CA.TI
Beta 1-Ca 1-Ti 3-Oxide	CA1O1.TI1O2	B_CA.TI
3-Ca 2-Ti 7-Oxide	CA3O3.TI2O4	CA3.TI2
4-Ca 3-Ti 10-Oxide	CA4O4.TI3O6	CA4.TI3
Calcium Metaborate	CA1O1.B2O3	CA.B2O3
Calcium Tetraborate	CA1O1.B4O6	CA.B4O6
Alpha 2-Ca 3-B 5-Oxide	CA2O2.B2O3	A_CA2.B2
Beta 2-Ca 2-B 5-Oxide	CA2O2.B2O3	B_CA2.B2
3-Ca 2-B 6-Oxide	CA3O3.B2O3	CA3.B2O3
Ca 2-Al 4-Oxide	CA1O1.AL2O3	CA.AL2O3
Ca 4-Al 7-Oxide	CA1O1.AL4O6	CA.AL4O6
12-Ca 14-Al 33-Oxide	CA12O12.AL14O21	CA.AL14
3-Ca 2-Al 6-Oxide	CA3O3.AL2O3	CA3.AL2
3-Ca 2-Al 6-O 6-Hydrate	CA3O3.AL2O3.H12O6	H.CA3.AL
4-Ca 2-Al 7-O 13-Hydrate	CA4O4.AL2O3.H26O13	H.CA4.AL
2-Ca Aluminosilicate	CA2O2.AL2O3.SI1O2	CA2.AL.S
Calcium Dimagneside	CA1MG2	CAMG2
Ca Magnesium Dioxide	CA1O1.MG1O1	CA.MGO

LONG NAME	FORMULA	SHORT NAME
Diopside	CA1O1.MG1O1.SI2O4	DIOPSIDE
Monticellite	CA1O1.MG1O1.SI1O2	MONTICEL
Akermanite	CA2O2.MG1O1.SI2O4	AKERMANI
Tremolite	CA2O2.MG5O5.SI8O16.HTREMOLIT 2O	
Merwinite	CA3O3.MG1O1.SI2O4	MERWINIT
Anorthite	CA1O1.AL2O3.SI2O4	ANORTHIT
Ca-Al Pyroxene	CA1O1.AL2O3.SI1O2	PYROXENE
Gehlenite	CA2O2.AL2O3.SI1O2	GEHLENIT
Grossular	CA3O3.AL2O3.SI3O6	GROSSULA
Sphene	CA1O1.TI1O2.SI1O2	SPHENE
2-Calcium Tin	CA2SN1	CA2SN
Calcium Tin	CA1SN1	CASN
2-Calcium Lead	CA2PB1	CA2PB
Calcium Lead	CA1PB1	CAPB
Calcium 2-Aluminum	CA1AL2	CA2-AL
Calcium 4-Aluminum	CA1AL4	CA4-AL
Calcium Zinc	CA1ZN1	CAZN
Calcium 2-Zinc	CA1ZN2	CA2-ZN
Calcium Uranate	CA1O1.U1O3	CAO.UO3
Calcium Hafnate	CA1O1.HF1O2	CAO.HFO2
Ca Molybdenum Tetraoxide	CA1O1.MO1O3	CAO.MOO3
Calcium Tungstate	CA1O1.W1O3	CAO.WO3
3-Ca Tungsten 6-Oxide	CA3O3.W1O3	CA3.WO3
Calcium 2-Iron 4-Oxide	CA1O1.FE2O3	CAO.FE2
2-Ca 2-Iron 5-Oxide	CA2O2.FE2O3	CA2.FE2
Calcium Metavanadate	CA1O1.V2O5	CAO.V2O5
Calcium Pyrovanadate	CA2O2.V2O5	CA2.V2
3-Ca Orthovanadate	CA3O3.V2O5	CA3.VE
Calcium Metazirconate	CA1O1.ZR1O2	CAO.ZRO2
Cadmium Fluoride	CD1F2	CDF2
Cadmium Chloride	CD1CL2	CDCL2
Cadmium Bromide	CD1BR2	CDBR2

LONG NAME	FORMULA	SHORT NAME
Alpha Cadmium Iodide	CD1I2	A_CD1I2
Beta Cadmium Iodide	CD1I2	B_CD1I2
Cadmium Oxide	CD1O1	CDO
Cadmium Sulfate	CD1S1O4	CDSO4
Cadmium Selenide	CD1SE1	CDSE
Cadmium Selenite	CD1SE1O3	CDSEO3
Cadmium Telluride	CD1TE1	CDTE
Cadmium Arsenide	CD3AS2	CD3AS2
Cd Monoantimonide	CD1SB1	CDSB
Cadmium Metasilicate	CD1O1.SI1O2	CDO.SIO2
Cadmium Aluminate	CD1O1.AL2O3	CDO.AL2O
Cd Gallium Tetraoxide	CD1O1.GA2O3	CDO.GA2O
Cadmium Tungstate	CD1O1.W1O3	CDO.WO3
Cadmium Titanate	CD1O1.TI1O2	CDO.TIO2
Gamma1 Cerium	CE	G1_CE
Gamma2 Cerium	CE	G2_CE
Delta Cerium	CE	D_CE
Cerium Dihydride	CE1H2	CEH2
Cerium Fluoride	CE1F3	CEF3
Cerium Chloride	CE1CL3	CECL3
Cerium Bromide	CE1BR3	CEBR3
Cerium Iodide	CE1I3	CEI3
Cerium Oxide	CE2O3	CE2O3
Cerium 1.72-Oxide	CE1O1.72	CEO1.72
Cerium 1.83-Oxide	CE1O1.83	CEO1.83
Cerium Dioxide	CE1O2	CEO2
Cerium Monosulfide	CE1S1	CES
Tricerium Tetrasulfide	CE3S4	CE3S4
Cerium Sulfide	CE2S3	CE2S3
Cerium Sulfate	CE2S3O12	CE2S3O12
Cerium Monotelluride	CE1TE1	CETE
Cerium Nitride	CE1N1	CEN
Dicerium Tricarbide	CE2C3	CE2C3

LONG NAME	FORMULA	SHORT NAME
Cerium Dicarbide	CE1C2	CEC2
Cerium 2-Silicon	CE1SI2	CESI2
Cerium Hexaboride	CE1B6	CEB6
Cerium 2-Aluminum	CE1AL2	CEAL2
Cerium 4-Aluminum	CE1AL4	CEAL4
Monocerium Aluminate	CE1AL1O3	CEALO3
Cobalt Difluoride	CO1F2	COF2
Cobalt Trifluoride	CO1F3	COF3
Cobalt Dibromide	CO1BR2	COBR2
Cobalt Diiodide	CO1I2	COI2
Cobalt Disulfide	CO1S2	COS2
Cobalt Selenite	CO1SE1O3	COSEO3
Tricobalt Nitride	CO3N1	CO3N
Dicobalt Phosphide	CO2P1	CO2P
Cobalt Monophosphide	CO1P1	COP
Cobalt Triphosphide	CO1P3	COP3
Cobalt 0.98-Antimonide	CO1SB0.98	COSB0.98
Cobalt Diantimonide	CO1SB2	COSB2
Cobalt Triantimonide	CO1SB3	COSB3
Cobalt Orthosilicate	CO2O2.SI1O2	SI.CO2O2
Cobalt Titanate	CO1O1.TI1O2	COO.TIO2
2-Cobalt Titanium 4-Oxide	CO2O2.TI1O2	TI.CO2O2
Dicobalt Boride	CO2B1	CO2B
Cobalt Monoboride	CO1B1	COB
Cobalt Tin	CO1SN1	COSN
Cobalt Aluminum	CO1AL1	COAL
2-Cobalt 5-Aluminum	CO2AL5	CO2AL5
Cobalt 3-Aluminum	CO1AL3	COAL3
2-Cobalt 9-Aluminum	CO2AL9	CO2AL9
Alpha Cobalt Tungstate	CO1O1.W1O3	A_COO.WO
Beta Cobalt Tungstate	CO1O1.W1O3	B_COO.WO
Cobalt 2-Iron 4-Oxide1	CO1O1.FE2O3	1_COFE2O
Cobalt 2-Iron 4-Oxide2	CO1O1.FE2O3	2_COFE2O

LONG NAME	FORMULA	SHORT NAME
Cobalt 2-Iron 4-Oxide2	CO1O1.FE2O3	3_COFE2O
Cobalt 2-Iron 4-Oxide3	CO1O1.FE2O3	4_COFE2O
Co Dichromium Tetraoxide	CO1O1.CR2O3	COO.CR2O
Cobalt Hydroxyd	CO_O1O2H2	COO(OH)2
Chromium1	CR1	CR_1
Chromium2	CR1	CR_2
Chromium Dichloride	CR1CL2	CRCL2
Chromium Trichloride	CR1CL3	CRCL3
Chromium(3) Oxide	CR2O3	CR2O3
Chromium Dioxide	CRO2	CRO2
5-Chromium 12-Oxide	CR5O12	CR5O12
8-Chromium 21-Oxide	CR8O21	CR8O21
Chromium Trioxide	CR1O3	CRO3
Chromium Sulfate	CR2S3O12	CR2S3O12
Copper(1) Sulfate	CU2S1O4	CU2SO4
Copper(2) Sulfate	CU1S1O4	CUSO4
Iron(2) Chloride	FE1CL2	FECL2
Iron(3) Chloride	FE1CL3	FECL3
Iron Oxychloride	FE1O1CL1	FEOCL
Wuestite1	FE0.947O1	1_WUESTI
Wuestite2	FE0.947O1	2_WUESTI
Iron(2) Oxide	FE1O1	FEO
Iron(2,3)Oxide1(Magnetit	FE3O4	1_FE3O4
Iron(2,3)Oxide2(Magnetit	FE3O4	2_FE3O4
Iron Trioxide Hydrate	FE2O3H2O1	H.FE2O3
Iron(2) Sulfate	FE1S1O4	FESO4
Iron(3) Sulfate	FE2S3O12	FE2S3O12
Magnesium Chloride	MG1CL2	MGCL2
Magnesium Hydroxide Chlo	MG1O1H1CL1	MGOHCL
Magnesium Sulfate	MG1S1O4	MGSO4
Magnesium Nitrate	MG1N2O6	MGN2O6
Manganese(2) Chloride	MN1CL2	MNCL2
Manganese(2,3) Oxide1	MN3O4	MN3O4

LONG NAME	FORMULA	SHORT NAME
Manganese(2,3)Oxide2	MN3O4	MN3O4
Manganese(3) Oxide	MN2O3	MN2O3
Manganese(2) Sulfate	MN1S1O4	MNSO4
Nickel Chloride	NI1CL2	NI1CL2
Nickel Monoxide1	NI1O1	1_NIO
Nickel Monoxide2	NI1O1	2_NIO
Nickel Monoxide3	NI1O1	3_NIO
Lead Chloride	PB1CL2	PBCL2
Trilead Tetraoxide	PB3O4	PB3O4
Lead Dioxide	PB1O2	PBO2
Lead Sulfate1	PB1S1O4	1_PBSO4
Lead Sulfate2	PB1S1O4	2_PBSO5
Lead Carbonate	PB1C1O3	PBCO3
Vanadium Dichloride	V1CL2	VCL2
Vanadium Trichloride	V1CL3	VCL3
Vanadium Monoxide	V1O1	VO
Vanadium Trioxide	V2O3	V2O3
Vanadium Dioxide1	V1O2	1_VO2
Vanadium Dioxide2	V1O2	2_VO2
Vanadium Pentoxide	V2O5	V2O5
Tungsten2	W1	W_2
Tungsten1	W1	W_1
Tungsten Dichloride	W1CL2	WCL2
Tungsten Tetrachloride	W1CL4	WCL4
Tungsten Pentachloride	W1CL5	WCL5
Tungsten Hexachloride1	W1CL6	WCL6
Tungsten Hexachloride2	W1CL6	WCL6
Tungsten Hexachloride3	W1CL6	WCL6
Tungsten Dioxide Dichlor	W1O2CL2	WO2CL2
Tungsten Monoxide Tetrac	W1O1CL4	WO1CL4
Tungsten Dioxide	W1O2	WO2
Tungsten 2.72-Oxide	W1O2.72	WO2.72
Tungsten 2.90-Oxide	W1O2.90	WO2.90

LONG NAME	FORMULA	SHORT NAME
Tungsten 2.96-Oxide	W1O2.96	WO2.96
Tungsten Trioxide1	W1O3	WO3
Tungsten Trioxide2	W1O3	WO3
Tungsten Carbonyl	W1C6O6	WC6O6
Zinc Chloride	ZN1CL2	ZNCL2
Zinc Sulfate1	ZN1S1O4	ZNSO4
Zinc Sulfate2	ZN1S1O4	ZNSO4
Zinc Sulfate Monohydrate	ZN1S1O4H2O1	H.ZNSO4
Zinc Sulfate Dihydrate	ZN1S1O4H4O2	2H.ZNSO4
Zinc Sulfate Hexahydrate	ZN1S1O4H12O6	6H.ZNSO4
Zinc Sulfate Heptahydrate	ZN1S1O4H14O7	7H.ZNSO4
Zinc Oxide Sulfate	ZN1O1ZN1S1O4	ZNOZNSO4
Zirconium Dichloride	ZR1CL2	ZRCL2
Zirconium Trichloride	ZR1CL3	ZRCL3
Zirconium Tetrachloride	ZR1CL4	ZRCL4
Zirconium Dioxide1	ZR1O2	1_ZRO2
Zirconium Dioxide2	ZR1O2	2_ZRO2
Cesium	CS	CS
Cesium Chloride1	CS1CL1	CSCL
Cesium Oxide	CS2O1	CS2O
Cesium Trioxide	CS2O3	CS2O3
Cesium Dioxide	CS1O2	CSO2
Cesium Sulfate1	CS2S1O4	1_CS2SO4
Cesium Sulfate2	CS2S1O4	2_CS2SO4
Cesium Carbonate	CS2C1O3	CS2CO3
Gallium	GA1	GA
Gallium Chloride	GA1CL3	GACL3
Gallium Oxide	GA2O3	GA2O3
Germanium1	GE1	GE_1
Germanium2	GE1	GE_2
Germanium3	GE1	GE_3
Germanium Tetrachloride	GE1CL4	GECL4
Germanium Dioxide1(Tetra	GE1O2	1_GEO2

LONG NAME	FORMULA	SHORT NAME
Germanium Dioxide2(Tetra	GE1O2	2_GEO2
Germanium Dioxide (Hexag	GE1O2	3_GE1O2
Hafnium1	HF1	1_HF
Hafnium2	HF1	2_HF
Hafnium Dichloride	HF1CL2	HFCL2
Hafnium Trichloride	HF1CL3	HFCL3
Hafnium Tetrachloride	HF1CL4	HFCL4
Hafnium Dioxide1	HF1O2	HFO2
Hafnium Dioxide2	HF1O2	HFO2
Mercury(1) Chloride	HG2CL2	HG2CL2
Mercury(2) Chloride	HG1CL2	HGCL2
Mercury(1) Sulfate	HG2S1O4	HG2SO4
Mercury(2) Sulfate	HG1S1O4	HGSO4
Indium	IN1	IN
Indium Monochloride1	IN1CL1	INCL
Indium Monochloride2	IN1CL1	INCL
Indium Dichloride	IN1CL2	INCL2
Indium Trichloride	IN1CL3	INCL3
Indium Oxide1	IN2O3	IN2O3
Indium Oxide2	IN2O3	IN2O3
Indium Sulfate	IN2S3O12	IN2S3O12
Iridium1	IR	1_IR
Iridium2	IR1	2_IR
Iridium Trichloride	IR1CL3	IRCL3
Iridium Dioxide	IR1O2	IRO2
Palladium1	PD1	PD_1
Plladium2	PD1	PD_2
Palladium3	PD1	PD_3
Palladium Chloride	PD1CL2	PDCL2
Palladium Oxide	PD1O1	PDO
Praseodymium1	PR1	PR_1
Praseodymium2	PR1	PR_2
Praseodymium3	PR1	PR_3

LONG NAME	FORMULA	SHORT NAME
Praseodymium Chloride	PR1CL3	PRCL3
Praseodymium Oxide	PR2O3	PR2O3
Praseodymium 1.72-Oxide	PR1O1.72	PRO1.72
Praseodymium 1.83-Oxide	PR1O1.83	PRO1.83
Praseodymium Dioxide	PR1O2	PRO2
Platinum	PT1	PT
Platinum Dichloride	PT1CL2	PTCL2
Platinum Trichloride	PT1CL3	PTCL3
Platinum Tetrachloride	PT1CL4	PTCL4
Potassium	K1	K
Potassium Chloride	K1CL1	KCL
Potassium Oxide	K2O1	K2O
Potassium Peroxyde	K2O2	K2O2
Potassium Dioxide	K1O2	KO2
Alpha Potassium Sulfate	K2S1O4	A_K2SO4
Beta Potassium Sulfate	K2S1O4	B_K2SO4
Alpha Potassium Nitrate	K1N1O3	A_KNO3
Beta Potassium Nitrate	K1N1O3	B_KNO3
Potassium Carbonate	K2C1O3	K2CO3
Cesium Chloride2	CS1CL1	CSCL
Lithium	LI1	LI
Lithium Chloride	LI1CL1	LICL
Dilithium Monoxide	LI2O1	LI2O
Lithium Peroxide	LI2O2	LI2O2
Lithium Sulfate1	LI2S1O4	1_LI2SO4
Lithium Sulfate2	LI2S1O4	2_LI2SO4
Lithium Carbonate1	LI2C1O3	1_LI2CO3
Lithium Carbonate2	LI2C1O3	2_LI2CO3
Lithium Carbonate3	LI2C1O3	3_LI2CO3
Rubidium	RB1	RB
Rubidium Chloride	RB1CL1	RBCL
Rubidium Oxide	RB2O1	RB2O
Rubidium Carbonate	RB2C1O3	RB2C1O3

LONG NAME	FORMULA	SHORT NAME
Titanium1	Ti1	Ti_1
Titanium2	Ti1	Ti_2
Titanium Dichloride	Ti1CL2	TiCL2
Titanium Trichloride	Ti1CL3	TiCL3
Titanium Monoxide1	Ti1O1	1_TiO
Titanium Monoxide2	Ti1O1	2_TiO
Dititanium Trioxide1	Ti2O3	1_Ti2O3
Dititanium Trioxide2	Ti2O3	2_Ti2O3
Trititanium Pentoxide1	Ti3O5	1_Ti3O5
Trititanium Pentoxide2	Ti3O5	2_Ti3O5
Titanium Dioxide(Rutile)	Ti1O2	RUTILE
Titanium Dioxide(Anatase)	Ti1O2	ANATASE
Thallium1	Tl1	Tl_1
Thallium2	Tl1	Tl_2
Thallium Chloride	Tl1CL1	TlCL
Thallium Trichloride	Tl1CL3	TlCL3
Thallium Oxide	Tl2O1	Tl2O
Dithallium Trioxide	Tl2O3	Tl2O3
Thallium Sulfate1	Tl2S1O4	1_Tl2SO4
Thallium Sulfate2	Tl2S1O4	2_Tl2SO4
Thulium1	Tm1	Tm_1
Thulium2	Tm1	Tm_2
Thulium Chloride	Tm1CL3	TmCL3
Thulium Oxide	Tm2O3	Tm2O3
Uranium1	U1	U_1
Uranium2	U1	U_2
Uranium3	U1	U_3
Uranium Trichloride	U1CL3	UCL3
Uranium Tetrachloride	U1CL4	UCL4
Uranium Pentachloride	U1CL5	UCL5
Uranium Hexachloride	U1CL6	UCL6
Uranium Dioxide	U1O2	UO2
4-Uranium 9-Oxide	U4O9	U4O9

LONG NAME	FORMULA	SHORT NAME
3-Uranium 8-Oxide	U3O8	U3O8
Uranium Trioxide	U1O3	UO3
Uranium Trioxide Monohyd	U1O3H2O1	UO3H2O
Uranium Disulfate	U1S2O8	US2O8
Sodium Chloride	NA1CL1	NACL

SECTION 9: USER-DEFINED COMPOUNDS

9.1 STARTING NEWCOMP.EXE

NewComp.exe is used to create new compounds that can be used in your VALI models.

The application NewComp.exe is located in the directory \bin of Belsim installation. It can be started directly from vali3 folder of the windows Taskbar.

User-defined compounds properties are stored in a user library called **user.bls** while the list of compounds is stored in the file **user.all**. Both files are located in the directory \usr of Belsim installation.

These two files should be copied and restored in the \usr directory at each new installation of Belsim software.

When delivered to the client, the user library contains one compound named H2O_USER that is simply a copy of the compound H2O from Belsim library.

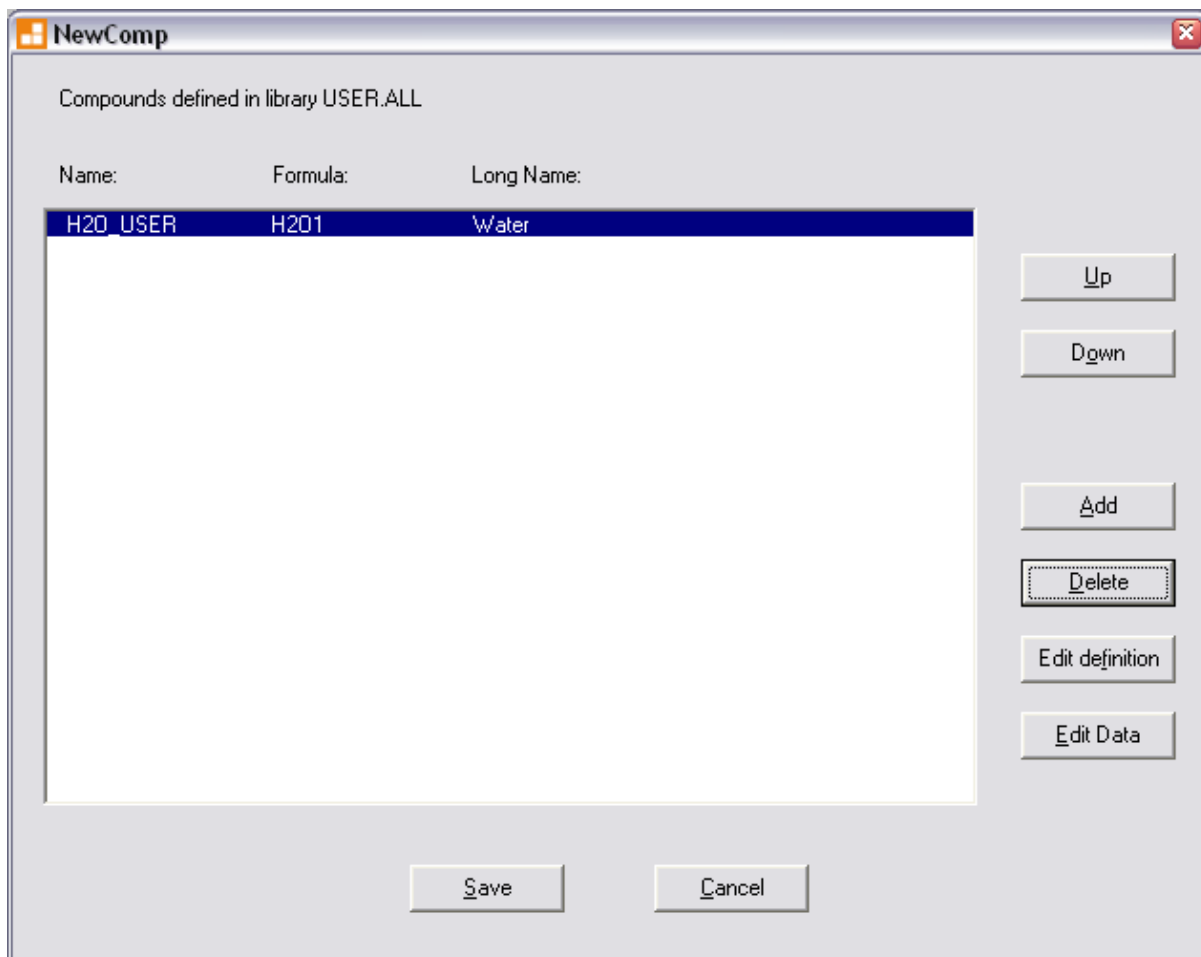


FIGURE 4 - THE NEWCOMP MAIN WINDOW

9.2 ADDING A NEW COMPOUND

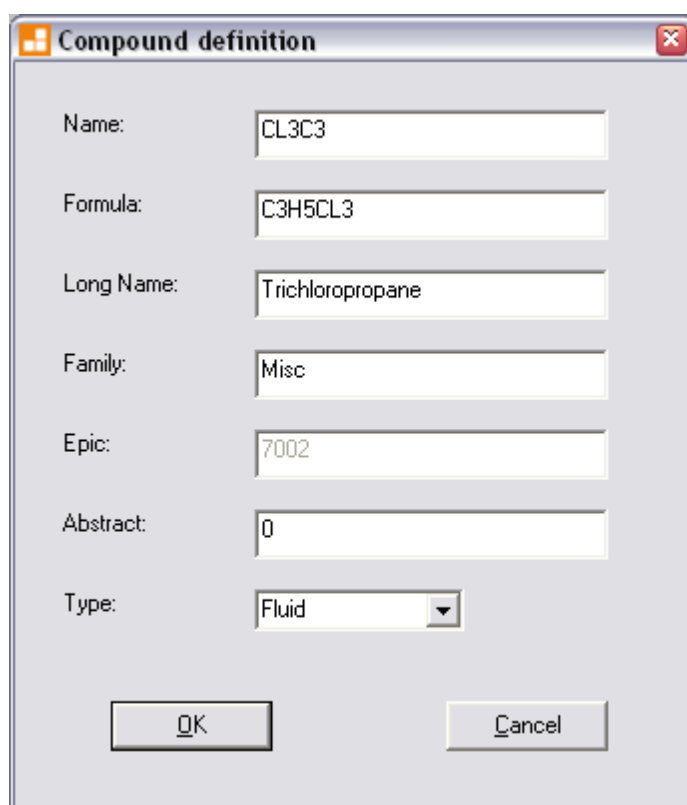
To add a compound:

- press the Add button.
- fill in the form :
 - Name: short name of the compounds in 8 characters maximum, without any space
 - Formula: compact chemical formula listing the total amounts of each element (ex: C₃H₅CL₃)
 - Long Name: this long name is used for documentation only.
 - Family: free field used to categorize the compounds when displaying the list of compounds in ValiModeller.
 - EPIC: internal index number.
 - Abstract number: introduced as a single number of 8 digits maximum containing only numbers from 0 to 9.
 - Type: Fluid or Solid
- press the OK button

Attention:

1. The modifications will only be made permanent after the OK key has been hit in the main notebook of the application.
2. The compound will be added after the selected compound in the main notebook
3. The compounds can be moved up and down with the [Up] and [Down] buttons.

Example for the trichloropropane:



Compound definition

Name: CL3C3

Formula: C3H5CL3

Long Name: Trichloropropane

Family: Misc

Epic: 7002

Abstract: 0

Type: Fluid

OK Cancel

FIGURE 5 - ADDING A NEW COMPOUND

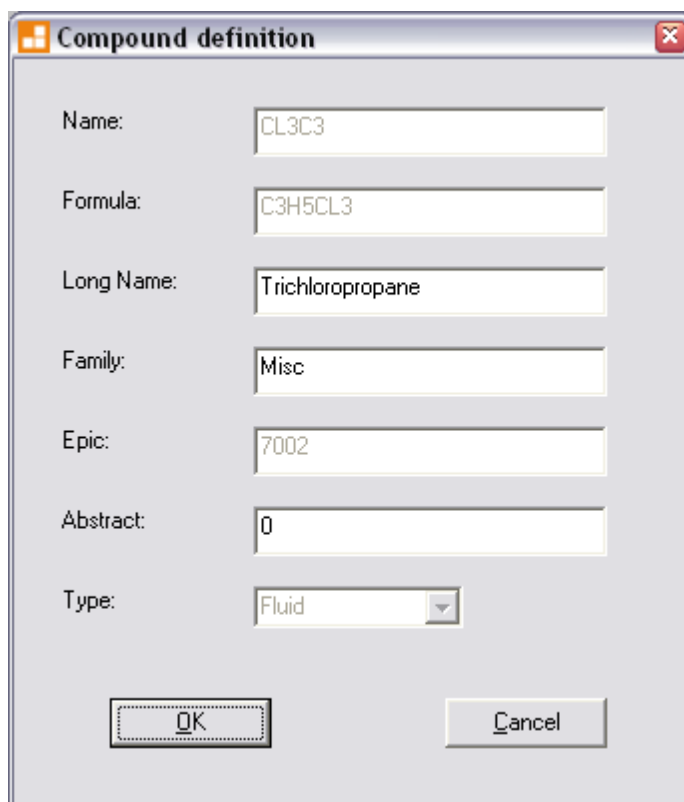
9.3 DELETING A COMPOUND

To delete a compound

- select the compound to be deleted from the list of substances of the main notebook
- press the **Delete** button

9.4 EDITING COMPOUND DEFINITION

Edit definition allows the user to change the Long Name, the Family and the Abstract Number of the selected compound. The other fields are not accessible once the compound has been created.



Compound definition

Name: CL3C3

Formula: C3H5CL3

Long Name: Trichloropropane

Family: Misc

Epic: 7002

Abstract: 0

Type: Fluid

OK Cancel

FIGURE 6 - EDITING THE COMPOUND DEFINITION

9.5 EDIT COMPOUND DATA

Edit data allows the user to access the whole set of parameters used to define the behaviour of the selected compound. At this point, one has to mention that for mass balance only, the sole parameters required are:

- the chemical formula
- the molecular weight (automatically computed from the chemical formula)

To perform mass and energy balance, the user will have to adjust the parameters required by the thermodynamic methods he plans to use. At this time, there is no estimation method (based on the group contributions,...) embedded in this tool to approximate the parameters required to carry out energy balance.

The values provided by the **Rough Evaluation** come from simplistic correlations just based on the molecular weight.

In any case, these estimations are not to be used for mass and energy balance.

Compound Thermodynamic Properties

Name:	Value:	Description:
HOF	-1	Standard enthalpy of formation (kJ/kmol)
GOF	-1	Gibbs s std. energy of formation (kJ/kmol)
MOLW	147.431996464729	Molecular weight (kg/kmol)
TB	-1	Boiling temperature (K)
TC	-1	Critical temperature (K)
PC	-1	Critical pressure (bar)
VC	-1	Critical volume (m3/kmol)
OMEGA	-1	Pitzer's acentric factor (-)
CPLA	-1	Liquid Heat Capacity (kJ/kmol/K)
CPLB	-1	$CPL = A + B \cdot T + C \cdot T^2 + D \cdot T^3$
CPLC	-1	.
CPLD	-1	.
TMIN(PV)	-1	Temperature range for vapor pressure (K)
TMAX(PV)	-1	.
PVAPA	-1	Vapour pressure equation (bar)
PVAPB	-1	$\ln(PVAP) = A + B/(T+C) + D \cdot T + E \cdot T^2 + F \cdot \ln(T) + G \cdot T^H$
PVAPC	-1	.
PVAPD	-1	.
PVAPE	-1	.
PVAPF	-1	.
CPIDA	-1	Ideal gas heat capacity (kJ/kmol/K)
CPIDB	-1	$CPID = A + B \cdot T + C \cdot T^2 + D \cdot T^3$

Rough Evaluation OK Cancel

FIGURE 7 - EDITING THE COMPOUND DATA

SECTION 10: USING CO-COMPLIANT THERMODYNAMIC PLUGS

10.1 THE PRINCIPLES

VALI has a Cape-Open (CO) compliant thermo-socket that enables thermodynamic calculations to be carried-out through thermodynamic packages made available by other software vendors in the form of CO compliant thermo plugs.

The implementation of the thermodynamic socket is compliant with CO-THRM-1 Version 1.0.

The use of third party plugs has been made as transparent and easy to use as possible and is done in two steps:

- installing the thermodynamic package on the computer where VALI is running
- creating a CAPE_OPEN thermodynamic model that will reference that package

10.2 DEFINING A CAPE-OPEN COMPLIANT THERMOD IN THE MODEL FILE

To define the CO thermod:

- select the Edit | Add Thermod command
- enter the Thermod name and select the type CAPE-OPEN 1.0
- press OK

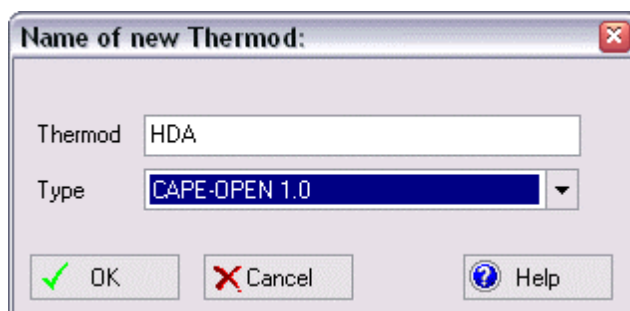


FIGURE 8 - CREATING A CO COMPLIANT THERMOD

Attention:

1. Compound names must unique across all Thermods (either native or Cape-Open compliant)
2. Compound names can not contain blank characters
3. You can mix streams with different Thermods through BBXVAL units. The correspondence between compounds is carried out according to their CAS number. When performing heat balances, take care that the same temperature and pressure is used as reference for enthalpy between all Thermods.

The list of Thermodynamic packages and systems that are installed on the computer is displayed.

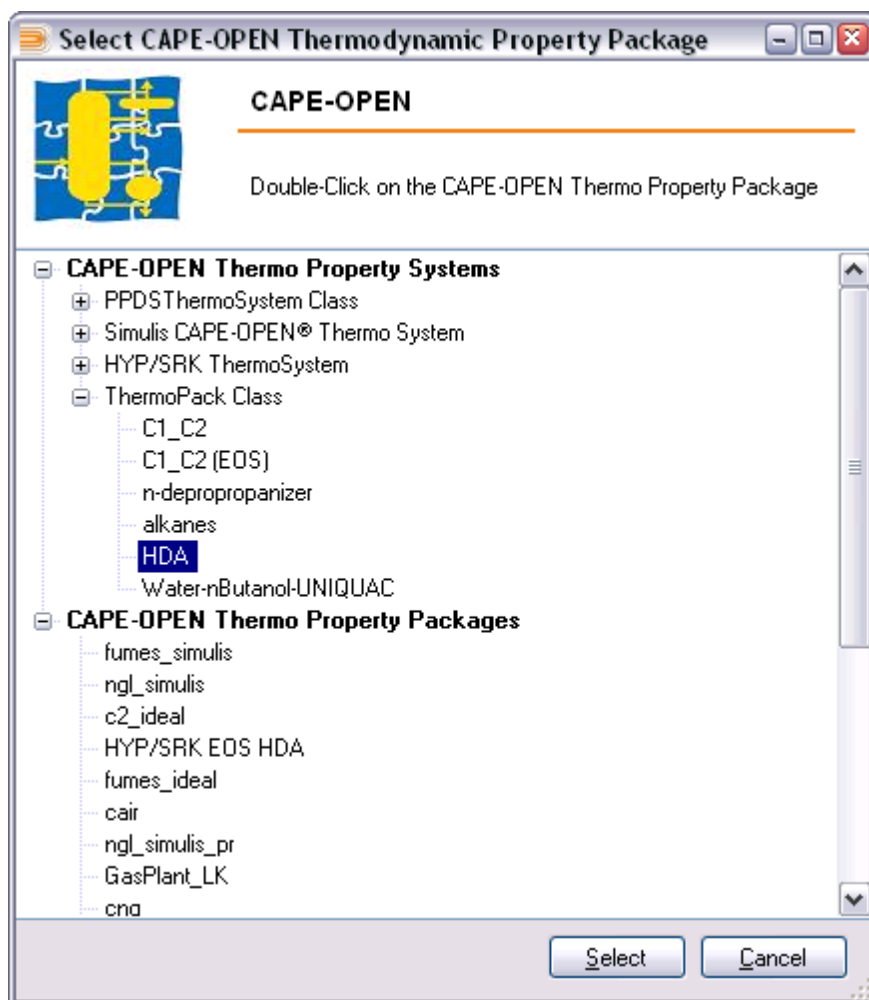


FIGURE 9 - SELECTING A CO COMPLIANT PACKAGE

Select the package you want to use and press Select.

10.3 REVIEWING THE PROPERTIES OF A CO COMPLIANT THERMOD

When editing a CO compliant Thermod, the ValiModeller native Thermod notebook is replaced by a window that displays the properties of the CO compliant package.

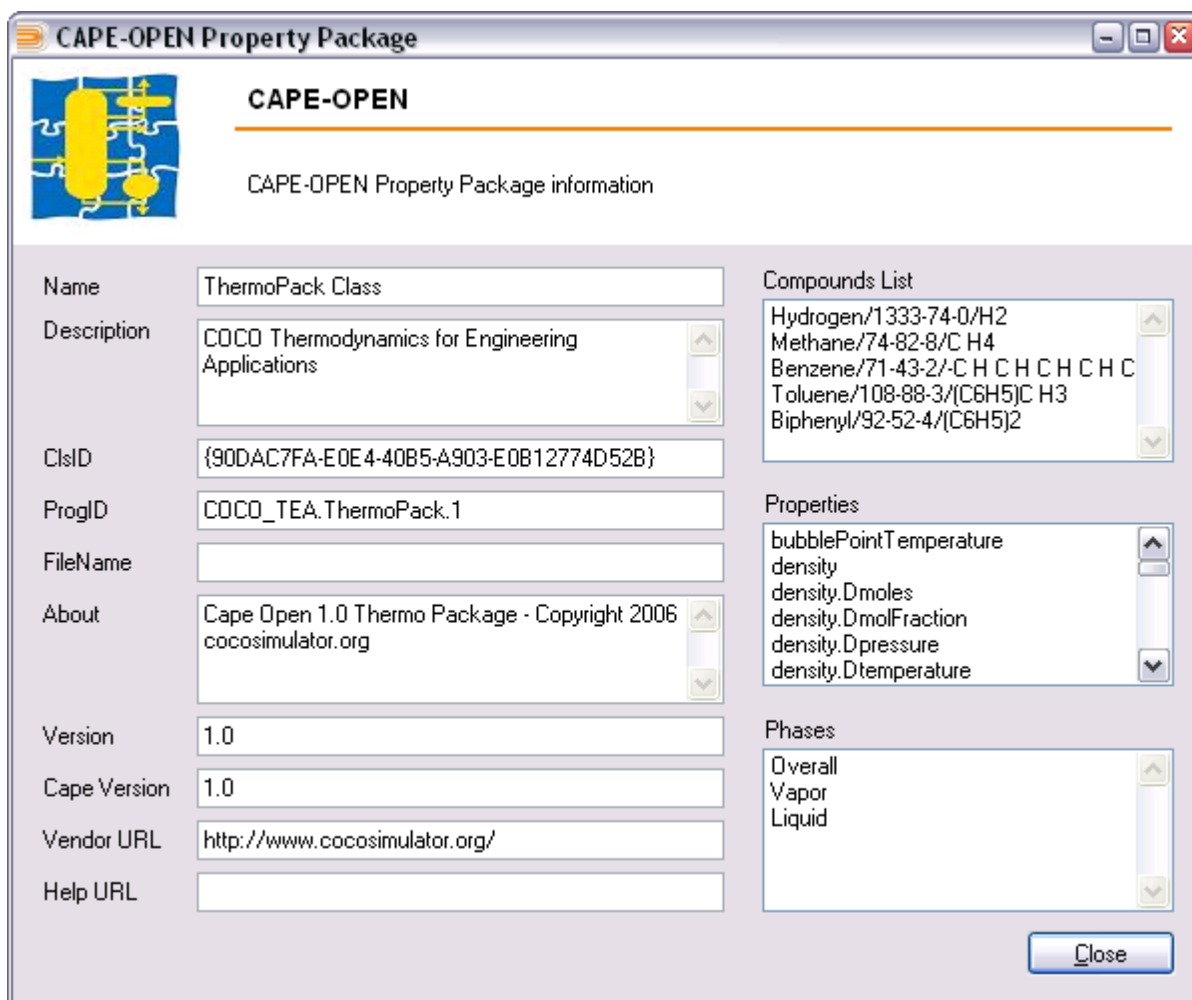


FIGURE 10 - PROPERTIES OF A CO COMPLIANT THERMOD

SECTION 11: UNCERTAINTY OF THERMODYNAMICS

11.1 INTRODUCTION

Physical characteristics and fluids properties are calculated through various thermodynamic methods which are subject to unmeasurable and hence random errors.

By default, these uncertainties are neglected by VALI. The parameter UNCTHER of the OPTVAL unit enables to activate these uncertainties.

At present time, such uncertainties are available and have been implemented for the steam tables (IAPWS methods).

11.2 CALCULATION PROCEDURE

When thermodynamics uncertainties are taken into account, the following procedure is applied:

- a first run is performed without thermodynamics uncertainties (the random errors are set to zero)
- the thermodynamics uncertainties are then relaxed and covariance factors are introduced between the deviations for points which are close to each other (in terms of p and T) and a second run is performed
- the reports are then generated on basis of the second run

For each link between stream variables and thermodynamic properties, an equation of the following form is generated (similarly to equation 91 of VDI-2048):

$$x = x(p, T) + DxTAB_y$$

where :
DxTAB_y is the unknown approximation deviation,
x is any thermodynamic property like enthalpy, entropy, volume,
y designates the phase type (L or V or S if property for saturation).

These variables will appear in the tables of :

- 'TAGS DEFINITION' with the initial definition of the tags
- 'CORRELATION FACTORS' with the values of correlation factors between tags
- 'TAGS OF THE SYSTEM' with the table of initial and validated values and accuracy

At present time, values for the uncertainties are only available for the IAPWS method. The standard deviations for H-S-V are defined from the values found in the table 43 p 40 of the "Release on the IAPWS Industrial Formulation 1997 for the Thermodynamic Properties of Water and Steam":

	REGION 1/3	REGION 2/3	REGION 2/5
Dv / %	0.002	0.007	0.001
Dh / (kJ kg ⁻¹)	0.014	0.073	0.012
Ds / (J kg ⁻¹ K ⁻¹)	0.022	0.094	0.025

Concerning the uncertainties on the saturated Pressure and Temperature, we have adopted values found in table 44 p 42 of the same document are used:

	T<623.15 K	623.15<T K
Dps / %	0.0033	0.0015
DTs / %	0.0003	0.0002

Attention: It can happen that some tags associated with thermodynamic uncertainties will be kept constant in the case they are associated with a non-existing phase. This can happen if a stream is defined as LV, but the final state is corresponding to pure L or V. In that case, only the variable associated with the existing phase will be relaxed and taken into account.

The calculation of correlation factors is based on the formula 102 on page 50 of the VDI 2048 guideline:

$$r_{\delta_1, \delta_k} = e^{-\left[\frac{(P_1 - P_k)^2}{k_p} + \frac{(t_1 - t_k)^2}{k_t} \right]}$$

The suitable values for the parameters are extracted from the reference [18] of the VDI 2048 guideline. The k_p and k_t parameters are found from the following relationships:

$$k_p = \frac{|\Delta P|}{\sqrt{\ln 2}}$$

$$k_t = \frac{|\Delta T|}{\sqrt{\ln 2}}$$

PHASE	DT / (K)	DP / (BAR)
Vapour	200	400
Liquid	200	10000

The correlation factors are generated between variables belonging to the same regions, there is never

correlation coefficients between measurements associated to different regions.

For this reason, it is impossible to have correlation factors between Liquid and Vapour properties for example. This is also not possible for saturation pressure if they are on both sides of the value of 623.15 K (which is the limit between regions 1 and 3).

The problem is then rerun with the values found without thermodynamic uncertainties as initial values.

The statistics about the size of the mathematical problem will be refreshed (except if there is no change) and the calculation of the a posteriori covariance matrix will start after identification of the active bounds which will be considered as additional constraints to the system.

Attention: If the second run is not successful, the solution found during the first run is restored and a special exitcode is issued.



belsim

CHAPTER 4

HOW TO BUILD YOUR DATA VALIDATION MODEL

SECTION 1: INTRODUCTION

Building a validation model requires the following information:

- a flow-sheet of the process you want to validate
- the location of measurements that are available including sample analyses (on-line and off-line)
- one (or many) sets of measurements
- although this is not strictly necessary, having a design heat and mass balance of the process is recommended for complex processes

The complete development of a model can be, somewhat arbitrarily divided in two parts:

- creating the model: defining the UNITS, STREAMS, measurements
- converging the model on a few cases and reviewing it: checking the model conformity to the actual process, adjusting the precision of the measurements, etc.

The creation of a model can be carried out in several ways. The sequence of operations proposed here is certainly one of the best methods when creating a new model. Other schemes are of course possible.

Better KISS than KICC: "Keep It Simple Stupid" is better than "Keep It Crazy Complicated".

Determine what you really want to validate and limit your model to this. Avoid complicating your model with unnecessary UNITS and STREAMS. If needed, you can easily add additional details afterwards.

Section 2 explains the different steps needed to build a validation model.

Sections 3 to 13 show how to use ValiModeller to build the model.

Section 14 shows a practical example of building a model.

SECTION 2: HOW TO BUILD A VALIDATION MODEL

2.1 PRELIMINARY WORK

The preliminary phase consists in answering some basic questions about the data validation model:

- what thermodynamic methods and chemical COMPOUNDS must be used ?
- what chemical REACTIONS have to be considered ?
- what validation UNITS will be used to model each equipment ?

Having answered these questions you can then set up the flow-sheet of the data validation model. This flow-sheet is generally similar to the process flow-sheet although some differences are sometimes observed. In some cases for example, several process UNITS can be modelled as one validation UNIT, whereas in some other cases, several validation UNITS will be necessary to model one process UNIT.

THE THERMODYNAMIC METHODS AND THE COMPOUNDS.

The choice of the compounds and thermodynamic methods depend on the scope of the data validation model.

REFINING, CHEMICAL AND PETROCHEMICAL PLANTS

For an **overall mass balance**, where only the total rates transferred between the UNITS is of importance, defining only one component (whatever it is) can be sufficient. The default thermodynamic methods will be sufficient.

For a **mass balance on a component per component basis**, the different chemical components must be defined. The default thermodynamic methods will be sufficient.

For a **heat and mass balance of a chemical plant**, the different chemical components must be defined. According to the system, and especially for strongly non-ideal systems, some more sophisticated thermodynamic methods might be necessary.

POWER PLANTS

For water and steam STREAMS (H₂O only), the IAPWS-IF97 method is recommended. The NBS or VDI method are also a good choice.

If combustion is involved, separate THERMODS should be used for the combustion air, the fuel and the fumes. Adequate THERMODS are available by default in the Empty model file (Empty.bls). Using default thermodynamic methods is usually sufficient for such STREAMS.

CHEMICAL REACTIONS

If chemical REACTIONS (including fuel combustion) are involved in the process, you can:

- either define the REACTION scheme by writing the different REACTIONS that can occur in the reactors and reference them in the UNIT (type RCTVAL, RLVVAL, RKPVAL, RKIVAL) modelling the reactor
- or allow any REACTION to occur, provided that atomic balances are satisfied. In such a case a UNIT of type BATVAL or BWHVAL must be used

In the first case, you must carefully select your REACTIONS, in such a way that on a given reactor, all REACTIONS are linearly independent. Failing to do so will induce singularities in the mathematical system and prohibit its solution.

SELECTING THE UNITS MODELS

The type of a UNIT determines what type of equipment is modeled and what equations are generated:

- clack-box/mixer:
 - BBXVAL: mixes up to 10 STREAMS and splits them into up to 10 STREAMS of different compositions
 - CUTVAL: breaks material or pressure loops of closed cycles
 - EFFVAL: mechanical/thermal STREAM efficiency box
 - NETVAL: network of pipes where STREAMS are mixed and split up on basis of a product code
- column:
 - COLVAL: saturation state of outlet STREAMS (saturated liquid or vapor) can be specified
 - DISVAL: tray by tray distillation column calculation model
 - CRUVAL: same as BBXVAL but petroleum fractions can be identified
- compressor:
 - COMVAL: includes isentropic efficiency equation
- controllers (user-equations):

- ATMOPRES: specifies atmospheric pressure (used a reference for pressure gauges)
- EQUVAL: equality of composition between 2 streams
- FLEXLIB: user-defined functions
- FLXVAL: user-defined equations
- OPXVAL: simple equations involving 2 variables
- VARVAL: additional variables
- heat exchanger:
 - ECOVAL: detailed simulation of a heat exchange section in an HRSG
 - EEXVAL: can model any heat exchange, no performance equation
 - HEXVAL: heat exchange between two STREAMS with performance equation
 - PRHVAL: heat exchange between a vapor that is condensed and a cold stream
 - SATVAL: allows you to set any saturation state on a STREAM
 - WALVAL: estimation of heat losses through walls of vessels
- pump:
 - PUMVAL: useful if one wants to identify pump power or if its power is known or related to another equipment (driven by a turbine). In many case, pumps are not modeled
- reactor:
 - BATVAL: any REACTION can occur (atomic mass balance)
 - BWHVAL: as BATVAL but with chemical equilibrium
 - CND-HNO3: Nitrous Gas Condenser
 - HEX-HNO3: Nitrous Gas Cooler
 - RCT-HNO3: NH3 Oxidation
 - RCTVAL: the possible REACTIONS are specified
 - RKIVAL: as RCTVAL but based on kinetics
 - RKPVAL: as RCTVAL but some reactions can be at (some distance of) chemical equilibrium
 - RLVVAL: as RCTVAL but with 2 outlet STREAMS in L/V equilibrium
- Separator:
 - LLEVAL: Liquid/Liquid equilibrium

- LLVVAL: Liquid/Liquid/Vapor equilibrium
- LVEVAL: Liquid/Vapor equilibrium
- WLEVAL: Water/Liquid equilibrium
- WLVAL: Water/Liquid/Vapor equilibrium
- splitter:
 - SPLVAL: splits a STREAM into up to 10 STREAMS of identical composition
- storage:
 - TNKVAL: mass, volume and/or composition balances of storage tank (involves accumulation and de-accumulation)
- turbine:
 - TURVAL: includes isentropic efficiency equation
- valves:
 - DPVAL: pressure drop in valves (can be function of rate)
 - PIPEVAL: pressure drop in line

2.2 DEFINING THE MODEL WITH VALIMODELLER

Defining a model involves the following steps:

STEP 1: CREATING OR SELECTING A MODEL FILE

The model file is a collection of models that are related. The way they are exactly related is up to the user. All models of a given process can be, for example, grouped together.

STEP 2: DEFINING THE THERMODS AND COMPOUNDS

When needed, select the additional chemical compounds and create the THERMODS that are necessary.

STEP 3: DEFINING THE CHEMICAL REACTIONS

Create the REACTIONS needed to describe your process.

STEP 4: DEFINING THE PROCESS : PFDS, UNITS, STREAMS

- Create new PFDs (Process Flow Diagrams)
- in each PFD: define the UNITS one by one and connect them with STREAMS. Review the various UNIT and STREAM parameters

STEP 5: DEFINING THE MEASUREMENTS

The measurements can be defined in two different ways:

- in the model file, through ValiModeller
- in a separate text file called a TAG file, which you handle through any text editor, like NotePad or WordPad. Using a spread-sheet application (like Excel, Lotus, etc.) is also pretty well indicated for such a task. You must however export the spread-sheet in a text form (ASCII file) so that ValiEngine can read it

When using a separate TAG file, the tags defined in the model are ignored.

SECTION 3: INTRODUCING VALIMODELLER

3.1 WINDOWS STANDARD

ValiModeller follows Windows standards for most if not all of its operations:

- click an object to select it
- double-click the object to Open it. Double clicking a PFD opens the corresponding flow-sheet, while double-clicking any other object (units, streams, thermods, etc.) opens its Notebook
- right-click an object to activate the Object menu, which presents the different operations you can perform with that object

Notebooks are associated with the following objects : PFDs, UNITS, STREAMS, THERMODS, REACTIONS, COMPOUNDS and Result Boxes. These Notebooks gather in different pages all the information of the corresponding object.

Notebooks are activated by double-clicking the object or by right-clicking the object and selecting the Data option in the Object menu.

The data are presented mainly through Tables. Cells with grey background are not editable. Only cells with white background are editable values.

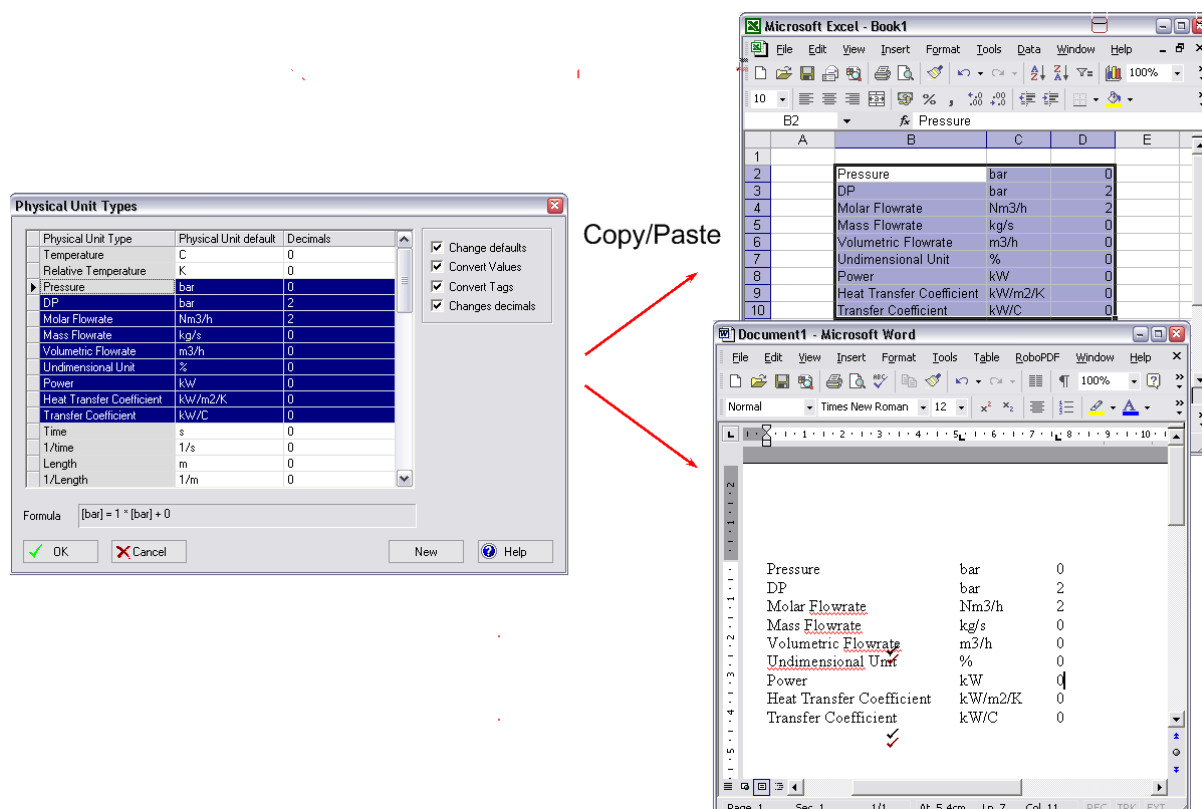


FIGURE 1 - COPY FROM TABLES TO CLIPBOARD

Tables are handled as follows

- use the arrow keys of your keyboard or mouse to navigate between the different cells of a table
- a little arrow on the right side of the active cell indicates a combo box
- to edit the contents of a cell, double-click on the cell or press "Enter" or the "Spacebar" key. Alternatively, you can directly enter a value to replace the existing value of in the cell
- using the mouse, you can select a range of cells or the entire table. Use CTRL-C to copy the selected range to the clipboard for exporting to external applications like Excel or Word

3.2 INFORMATION COHERENCY

With ValiModeller you can add new elements to your models but you can also review the data of existing elements. A word of caution is here necessary: the validated data of the objects are only coherent with respect to the last successful run of the model. Incoherence in the validated data can therefore result from the following causes:

- any change made to a model can result in data incoherence: addition or deletion of a UNIT or of a STREAM, but also change of any parameter of a UNIT or of a STREAM
- as ValiModeller allows you to run parts of models, the parts that have not been considered in the last run, are of course not necessarily coherent with the rest of the model

The UNITS and STREAMS data must therefore always be considered with great care.

After running ValiEngine (or when any other application has changed the model file), ValiModeller will propose:

- to reload the results: only the data are read from the file, assuming the model structure has not changed. This operation is quicker but might fail if some structural changes have been made to the model by the external application
- to take no action: the results are not loaded and the next run will restart from the initial point presented by ValiModeller

The last option allows you to first look at the results before loading them. Depending on these results you can decide later on to load the results back in ValiModeller and use the calculated results as a new initial point for the next run, through the command **Application | Load Results** (or its corresponding button ).

Attention:

1. When running ValiEngine, the results are saved in the file only if the run is successful.
2. Even if the run is successful, a runtime option allows you to suppress the saving of the results by ValiEngine.
3. In both cases, the **Application | Load results** command will restore the starting point as it was before the last run.

3.3 STARTING VALIMODELLER

To start ValiModeller, you can use one the following methods.

- use the ValiModeller option in the Belsim | Vali4 Programs Start Menu
- click on the corresponding Belsim model file in Windows Explorer

Alternatively, you can create a short-cut to ValiModeller.exe on your desktop.

If you started ValiModeller without specifying a model file, you will get an empty window.

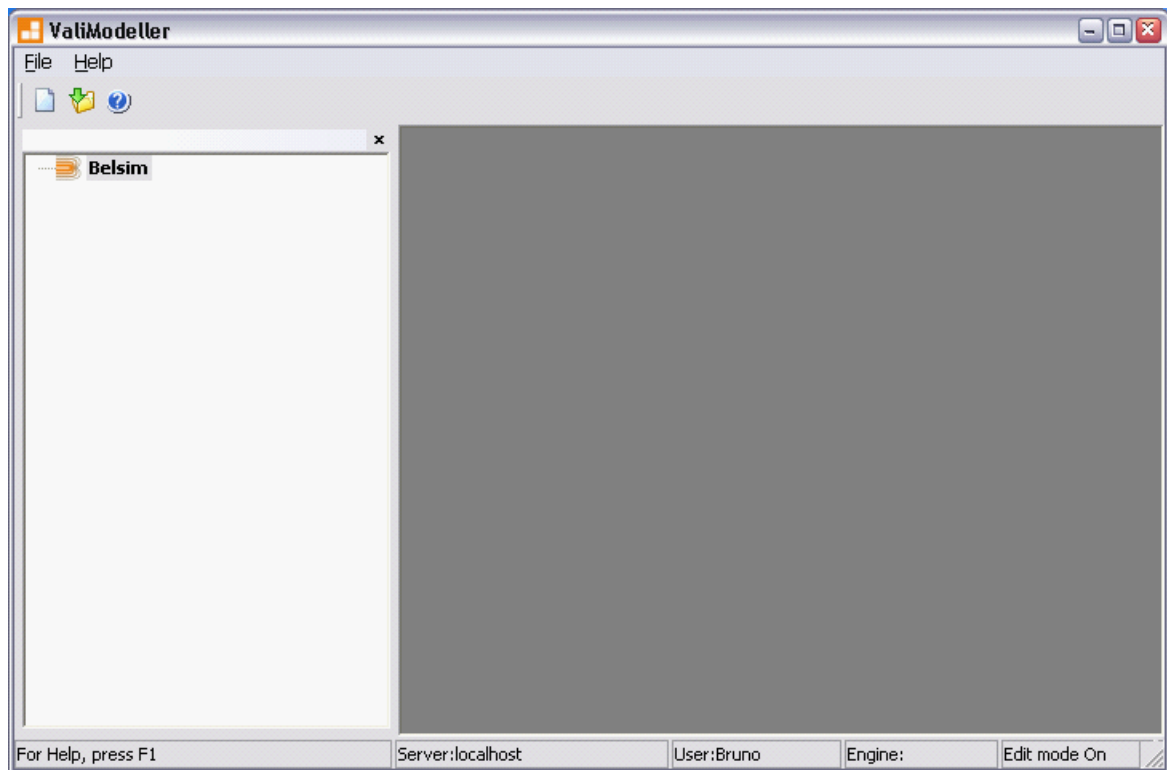


FIGURE 2 - VALIMODELLER EMPTY SCREEN

Use:

-  or **File | Open** to open an existing model
-  or **File | New** to create a new model

When opening an existing model, ValiModeller will open the main PFD (named by default MAIN) in View Mode.

Attention: if you upgraded from an earlier version of VALI, the model will be converted automatically when reading it.

3.4 THE BROWSE TREE

The Browse Tree can be activated or deactivated using the Browse Tree Button , the **Option | Browse Tree** Menu command or the short-cut Ctrl+B. The Browse Tree gives a global overview of the model structure. It is by default located on the left side of the screen but it can be moved as a toolbar anywhere in application area.

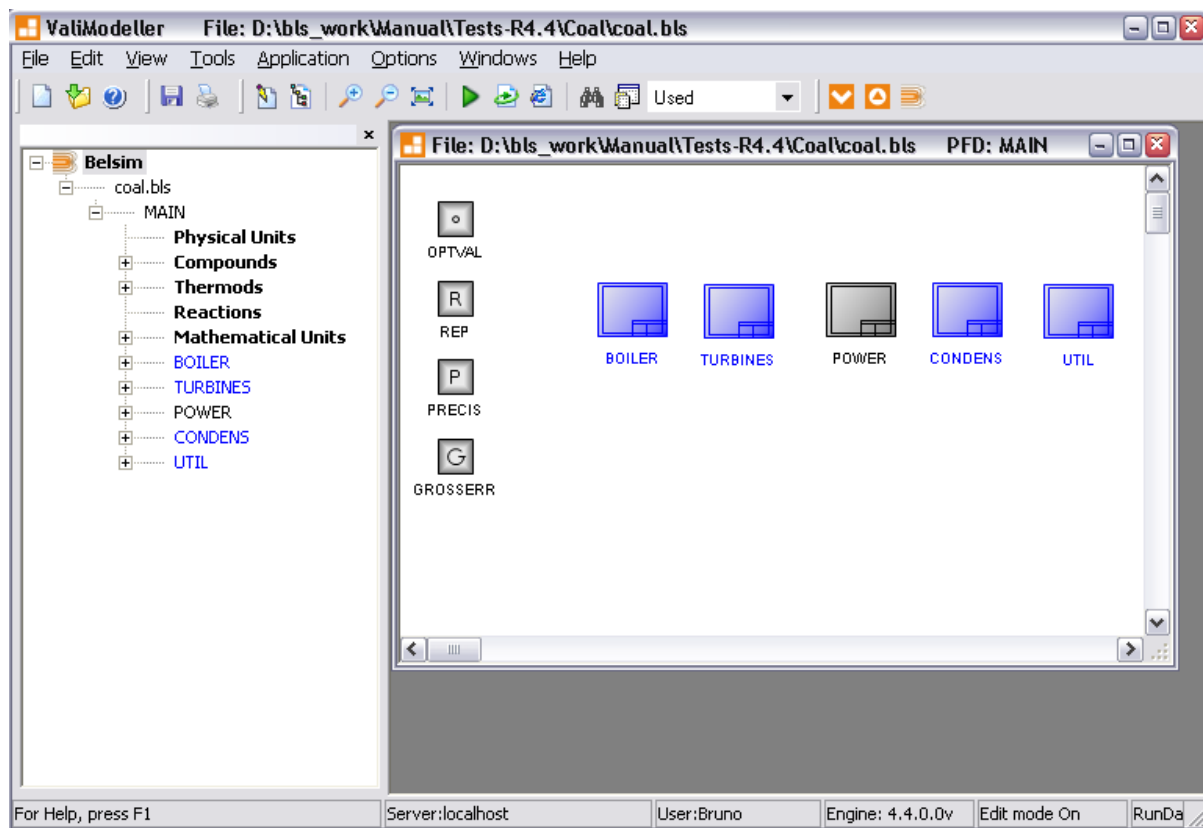


FIGURE 3 - VALIMODELLER MAIN PFD OPENED IN VIEW MODE

THE LEFT SIDE: BROWSE TREE

On the left side, appear successively, for each open model, the common objects (Physical units, COMPOUNDS, THERMODS, REACTIONS and Mathematical UNITS) followed by the list of PFDs.

Clicking on a  will expand the tree with more information about the corresponding item while clicking on a  will collapse this information. For example expanding a PFD will display its contents: UNITS, STREAMS and sub-PFDs.

Clicking on an PFD selects that PFD in the PFD View. **Clicking** on a UNIT or a STREAM, activates the corresponding object in the current PFD and highlights its symbol in yellow in the PFD View.

Double-clicking an object opens its Notebook, except for PFDs which are opened when double-clicking them.

Right-clicking an object opens its Object Menu.

THE RIGHT SIDE: PFD VIEW

Clicking on an PFD, STREAM or UNIT selects it.

Double-clicking a PFD opens it in the PFD View. **Double-clicking** a UNIT or a STREAM opens its Notebook.

Right-clicking an object opens its Object Menu.

THE STATUS BAR

The status bar presents the following information:

- Server: name of Belsim License Server
- User: name of the user
- Engine: version of ValiEngine used during last run
- Edit mode: status (ON/OFF) of the edit mode (see page 20)
- RunDate: date/time of the last dataset used
- Quality: Penalty/Chi2 of the last run
- Penalty: weighted sum of penalties of the last run
- PFD: name of the PFD used during the last run

3.5 MENU ITEMS

The menu bar presents the following items:

- File
 -  New: to create a new file
 -  Open: to open an existing file
 - Close File <file>: to close the current file (file)
 - Import Tag File: to import tags defined in a TAG file model
 -  Save: to save the current file
 - Save As: to save the current file under a new name
 -  Print...: to print the current PFD
 - Print to File...: to print the current PFD as an emf file (enhanced metafile)
 - Preview: to preview before printing
 - Print All Pfds: to print all PFDs of the current model
 - Print Setup...: to select the printer and some other printer settings
 - Fit to Page: toggle to Fit the printing to the page size. The drawing will be scaled so that it fits on a single page
 - Exit: to close the application
- Edit
 - Edit | Add PFD: to add a PFD (Process Flow Diagram)
 - Add Unit: to add a UNIT
 - Add Stream: to add a STREAM
 - Add Thermod: to add a THERMOD
 - Add Compound: to add a COMPOUND
 - Add Reaction: to add a REACTION
 - To Clipboard: to copy the active PFD to the clipboard
- View
 -  Zoom In: to zoom in the current PFD
 -  Zoom Out: to zoom out the current PFD
 -  Full Fit: to zoom out the current PFD until it fits the View area
 - Toolbar: to activate the toolbar

-  Find: to activate the FIND facility
- Hide Stream names: to hide the STREAM names
- Hide Unit names: to hide the UNIT names
- Hide Results: to hide the Result boxes
- Tools
 -  Measurements: to edit the values of the Tags defined in the model
 - Replace Tags Default Value: to replace all the tags default values by the measured values
 - Physical Units: to handle the Physical units
 - Plants: to handle the Plants
 - Highlight: to control the colour highlighting in function of tags
 - None: no colour highlighting
 - All Measurements: highlights all objects on which a Tag is defined
 - Active Measurements: highlights all objects on which a Tag is defined an active
 - Filtered: highlights all objects on which a Tag contains a filter activated by ValiFilter
 - Penalty: highlights all objects according to the penalty of its tags
 - Display: to display UNITS and STREAMS status and select which tags values are displayed in the notebooks (a combo-box is also available on the toolbar)
 - None: UNITS and streams status is not displayed
 - Default: display UNITS and STREAMS status as set from GUI for the next run
 - Measured: display UNITS and STREAMS status as set after mea files reading
 - Used: display UNITS and STREAMS status finally used by ValiEngine (including filtering, OFF propagation...)
- Application
 -  Calculate: to launch ValiEngine and perform the calculations
 -  LicenseManager: to launch Belsim LicenseManager
 -  ValiAnalytics: to launch ValiAnalytics
 - Results: to display ValiEngine Result files
 - Html Report: Complete html report

- Log File (r1v): Summary of the run with warning and error messages
- Runtime Messages (r2v): Messages that are usually sent to the screen (This file is created only when running with -R2S option (Don't show summary of events))
- Mathematical Details (r3v): List of equations and variables, trace of the convergence algorithm.
- Full Report (r4v): Main report file
- Suspected tags (r5v): List of measurements that are eliminated or flagged as suspicious
- Tags Report (r6v): Tags report and summary of the results
-  Load Results: to load the results of the last run
- Options
 -  Edit mode: to toggle the Edit mode
 -  Browse Tree: to toggle the Browse Tree
 - Add Multiple Objects: to toggle the Multiple UNITS/STREAMS creation option, allowing to create several UNITS and STREAMS in a row.
 - History: to activate the project history window
 - Configure: to edit other options
- Windows
 - Cascade: to cascade the already opened PFDs
 - Tile Horizontal: to tile horizontally the already opened PFDs
 - Tile Vertical: to tile vertically the already opened PFDs
 - Maximize current PFD: to maximize the current PFD
 - Minimize current PFD: to minimize the current PFD
 - Restore window: to restore current PFD to its previous size
 - <file> -> <PFD>: list of the opened PFDs
- Help
 -  Help Topics: to activate the Help facility
 - About ValiModeller: to display the About Belsim window

The toolbar provides instant access to the most useful commands.

3.6 EDIT MODE

ValiModeller can be operated in two modes: **Edit mode ON** or **Edit mode OFF**.

In **Edit mode OFF** you can review the model and update the UNITS and STREAMS data.. All operations on the structure of a model are disabled : creation or deletion of objects (PFDs, UNITS, STREAMS, REACTIONS, THERMODS, COMPOUNDS, Measurements, Result boxes) but also any operation related to the Graphical representation of the model: symbols of UNITS (including their position and size), drawing of STREAMS, etc. The background of PFDs is set in light grey.

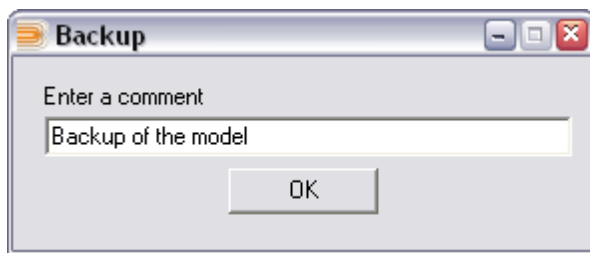
To enter any structural or graphical change to a model you must activate **Edit mode** through the **Options|Edit mode ON** command or the **Edit Mode** button . The background of PFDs is set in white.

Attention: when loading a model for which some tags are not registered in the LicenseServer or when loading a model which has been modified by a non-commercial license, the Edit Mode is set to OFF and cannot be reset to ON (Read Only Mode).

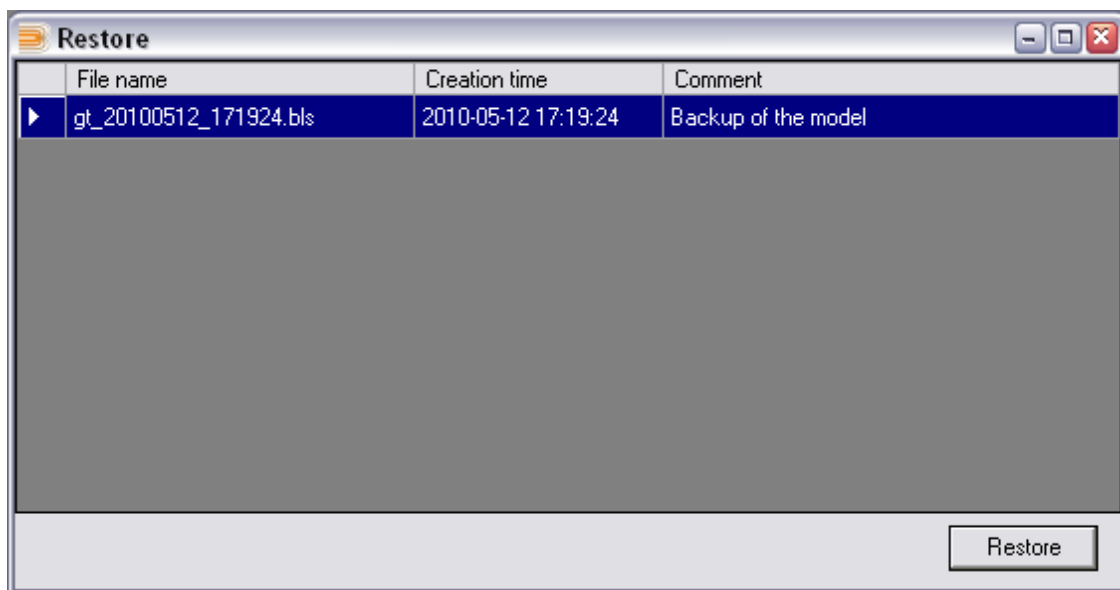
3.7 TOOLBAR AND KEYBOARD SHORT-CUTS

The toolbar presents the most useful commands of the menu:

-  New: to create a new file
-  Open: to open an existing file
-  Help Topics: to activate the Help facility
-  Save: to save the current file
-  Print...: to print the current PFD
-  Backup Model: to create a backup of the model (in ModelBackup directory). When clicking this button, a comment can be entered to describe the backup:



-  Restore Model: to restore a previously created back-up of the model. When clicking this button, the model is closed, and a window prompts for choice of a backup, displaying comments:



-  Edit mode: to toggle the Edit mode
-  Browse Tree: to toggle the Browse Tree
-  Zoom In: to zoom in the current PFD

-  Zoom Out: to zoom out the current PFD
-  Full Fit: to zoom out the current PFD until it fits the View area
-  Calculate: to launch ValiEngine and perform the calculations
-  Load Results: to load the results of the last run
-  Find: to activate the FIND facility
-  Measurements: to edit the values of the Tags defined in the model
-  LicenseManager: to launch Belsim LicenseManager
-  ValiAnalytics: to launch ValiAnalytics

Some keyboard short-cuts are also available:

- F1: opens the Help facility
- CTRL+F: opens the Find window (View | Find)
- CTRL+R: runs ValiEngine (Application | Calculate)
- CTRL+M: opens the View Measurements window (Tools | Measurements)
- CTRL+E: toggles Edit mode ON/OFF (Options | Edit Mode On/Off)
- CTRL+B: toggles Browse Tree ON/OFF (Options | Browse Tree On/Off)
- CTRL+O: opens an other file (File | Open)
- CTRL+P: calls the Print facility (PFD | Print)
- CTRL+S: saves current file (File | Save)

3.8 SHOWING UNIT AND STREAM NAMES AND RESULT BOXES

The **View | Hide STREAM Names** command allows you to suppress the display of these names on the flow-sheets. Similarly the command **View | Hide UNIT Names** allows to hide the UNIT Names in the flow sheet and the command **View | Hide Results** allows to hide the result boxes.

You can save these settings in your user profile through the **Options | Configure** notebook.

3.9 CHANGING CONFIGURATION OPTIONS

The **Option | Configure** command opens a notebook where various options can be edited in different pages:

- Options: various editing, display and printing options
- Prefix names: defines the default names for units and streams
- Model Button: user-defined buttons giving access to external applications
- Model Measurements: user-defined stream measurements

Attention: The settings of the first two pages can be saved in the user profile by clicking the Set as Defaults button so that they will be automatically used during the next session. The user settings are saved in a file called after the name of the user (<user>.ini) in the directory usr of Belsim installation.

OPTIONS PAGE

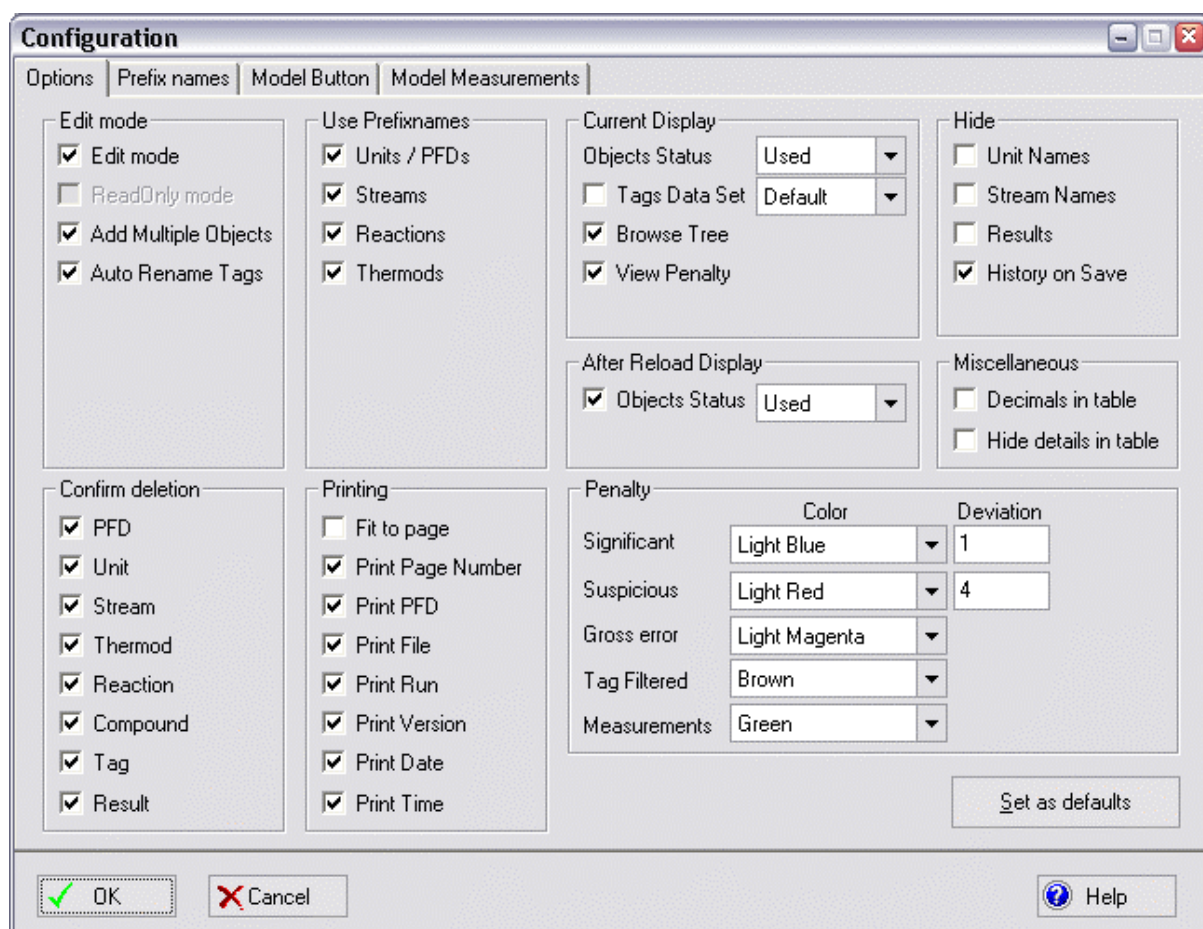


FIGURE 4 - THE OPTIONS PAGE OF THE CONFIGURATION NOTEBOOK

EDIT MODE

- Edit Mode: to enable/disable model edition
- Add Multiple Objects: to automatically add several UNITS or STREAMS in one shot
- Auto-Rename Tags: to automatically rename tags when renaming a unit or stream (provided the tag name is on basis of the object name followed by the a "_")

CONFIRM DELETION

You can specify here whether ValiModeller should ask a confirmation before actually deleting an object.

Attention: REACTIONS, THERMODS and COMPOUNDS can only be deleted if they are not referred by another object (e.g. you cannot delete a REACTION that appears in the list of REACTIONS of a UNIT).

USE PREFIX NAMES

When creating a new object, a name is created on basis of an optional prefix and of a number which is automatically incremented. The options of this section enable to specify whether or not to use the prefix.

For units and streams, a different prefix can be configured for each unit and stream type, as defined in the page **Prefix names** of the **Configuration** notebook.

For THERMODS and REACTIONS, the prefix is respectively THER and REAC.

PRINTING

You can specify here:

- whether the drawing should be scaled down in order to fit in on page
- which additional information should be included on the printed page

CURRENT DISPLAY

This part corresponds to the Menu commands **Tools|Highlight** and **Tools|Display**. It also enables to activate/deactivate the Browse Tree and the coloring of objects according to their penalty.

Object Status corresponds to the current display status:

- None: UNITS and STREAMS status is not displayed
- Default: display UNITS and STREAMS status as set from GUI for the next run
- Measured: display UNITS and STREAMS status as set after mea files reading
- Used: display UNITS and STREAMS status finally used by ValiEngine (including filtering, OFF propagation, etc.)

Tags Data Set option corresponds to the set of tags values shown when you open a notebook (Default, Measured or Used). If the check-box is not checked, this setting is ignored and the set of tags values is selected on basis of the Object Status setting.

AFTER RELOAD DISPLAY

This part corresponds to the Objects status shown after a reload of the file (typically after a run).

HIDE

These settings enable to hide:

- the UNIT and STREAM names in the PFDs
- the Result boxes in the PFDs
- the History window when saving a model

PENALTY

These settings specify which colors must be used for the tags highlighting system. You can also specify the threshold values for suspicious and significant deviations.

MISCELLANEOUS

You can specify here:

- values shown in the notebooks will be displayed with the number of decimals specified according to the physical units
- when checked, the details are hidden in the measurements window to expand the size of the measurements table

PREFIX NAMES PAGE

In the **Prefix Names** page you can control the **auto-naming** facility. Each unit and stream type can have its own prefix, which will be used as basis for generating the names at creation time.

Attention: You can click on the fields of the table header to change the sorting order of the list of units.

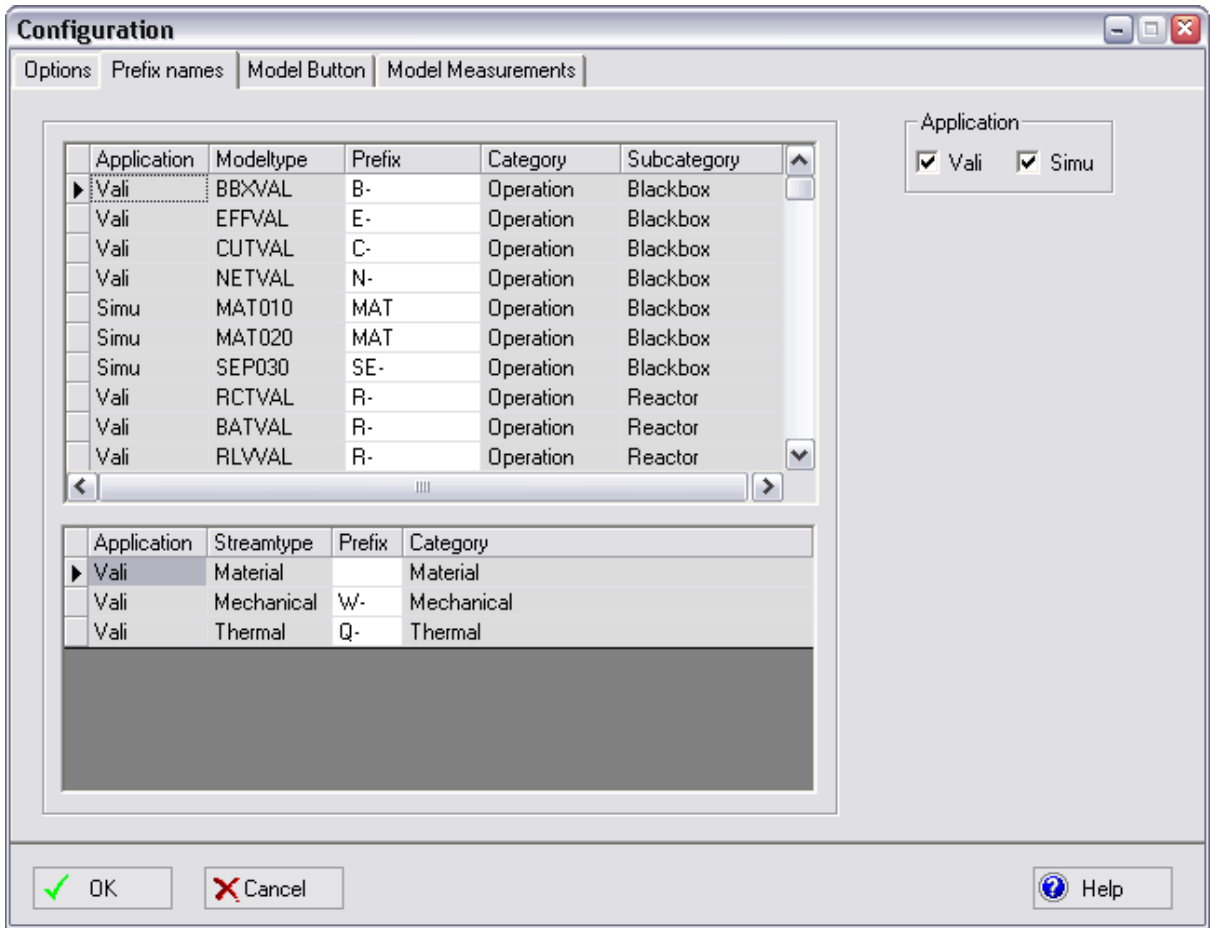


FIGURE 5 - THE PREFIX NAMES PAGE OF THE CONFIGURATION NOTEBOOK

MODEL BUTTON PAGE

The Model Button page defines personalized buttons visible in the toolbar allowing you to launch external applications. For each model, you can define up to 10 buttons as follows:

- Application path: path of the executable file to be launched when the button is clicked (mandatory)
- Start in: path of the directory where the external application will be launched. If not specified, the application will start in the directory of the application
- Icon: path of the ".ico" file displayed in the toolbar. If not specified, the Belsim icon is used
- ToolTip: text displayed when the cursor moves over the button
- Arguments: list of arguments that are used by the application. Clicking on the button "List" shows a list of predefined arguments
- Run Vali: when checked, ValiEngine is run before or/and after the application

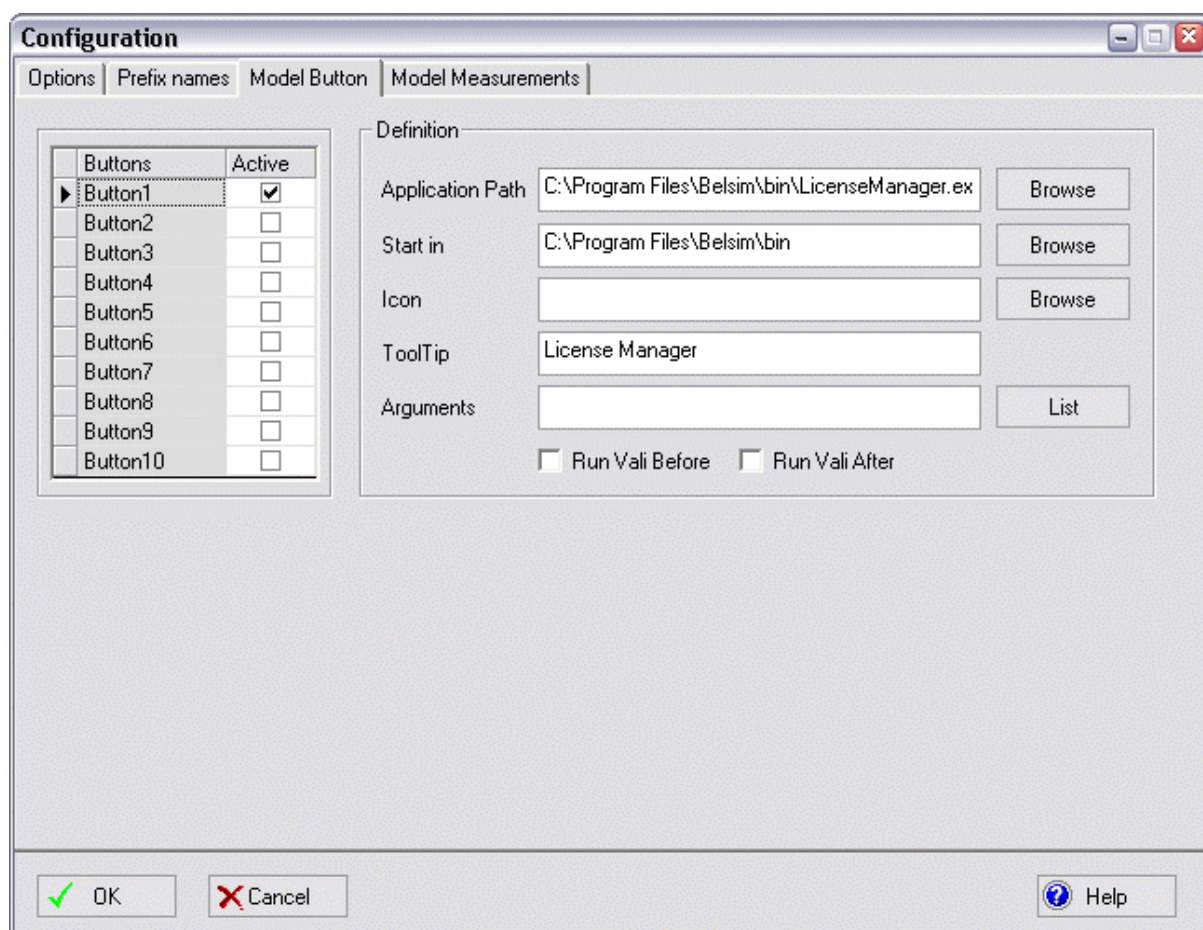


FIGURE 6 - THE MODEL BUTTON PAGE OF THE CONFIGURATION NOTEBOOK

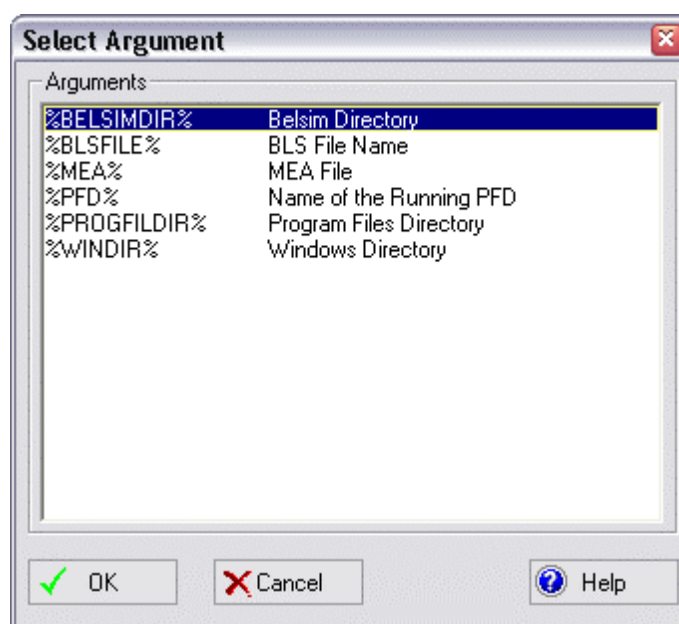


FIGURE 7 - THE LIST OF ARGUMENTS OF THE MODEL BUTTON PAGE

3.10 MODEL MEASUREMENTS PAGE

The Model Measurements page lists user-defined stream measurements. To create a new measurement, press Add button. Once a new measurement type is created, the model needs to be closed and reopened.

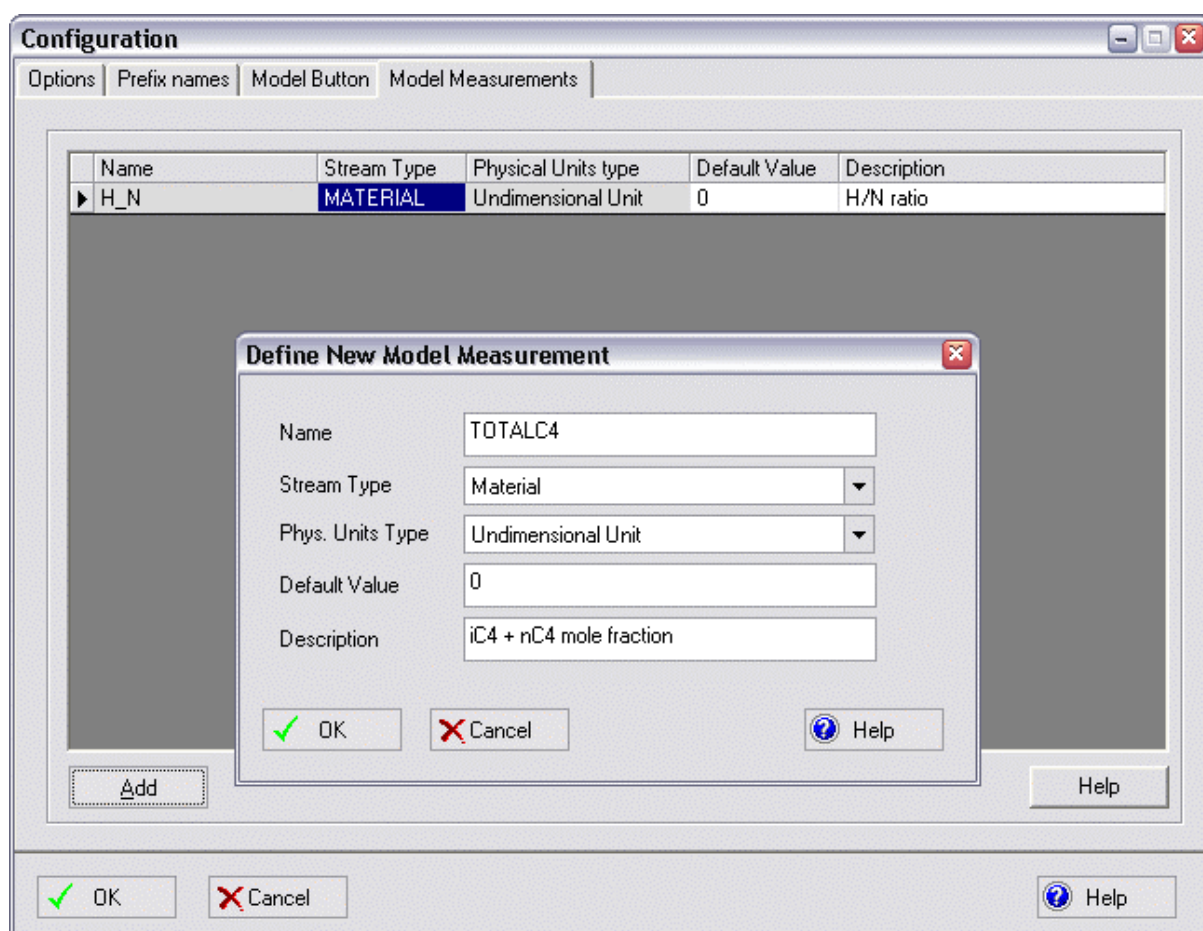


FIGURE 8 - USER STREAM MEASUREMENT TYPES

Once a user measurement is available in the model, it has to be defined using FLEX code.

3.11 THE FIND FACILITY

The **View|Find** command (or the  button) opens the **Find** window where you can list the PFDs, UNITS, STREAMS, TAGS, THERMODS, COMPOUNDS, REACTIONS and RESULT BOXES. The top field allows restricting the number of objects that are shown in the table. Clicking the **OK** button opens the notebook of the selected object.

If you deactivate the **Close on Ok** check box then the **Find** window will remain open after opening any object, until you press the **Cancel** button. Checking **Stay on top**, will maintain the window on top of the other windows.

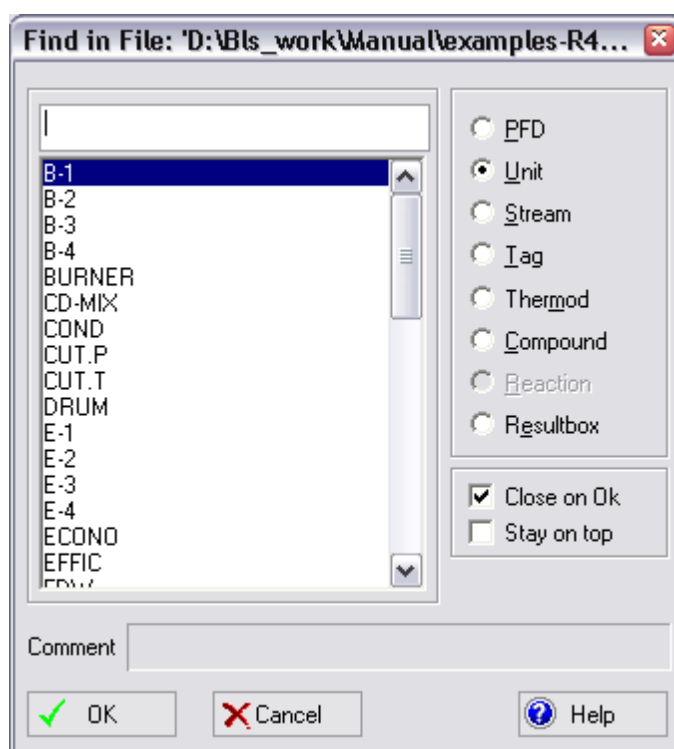


FIGURE 9 - THE FIND WINDOW

3.12 RENAMING OBJECTS

Renaming objects is only possible in Edit Mode. The Rename Object window can be activated through the Object Menu (right-click) of each object. It can also be activated via the Rename button of the notebooks.

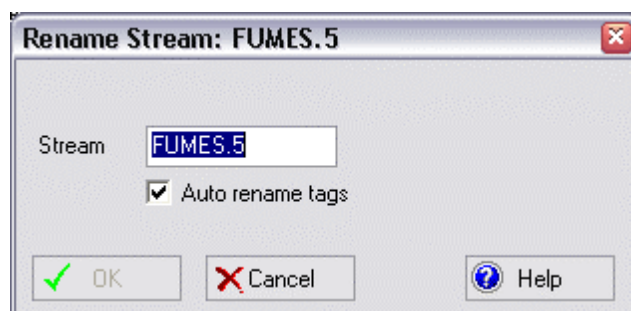


FIGURE 10 - RENAMING OBJECTS

For UNITS, STREAMS and COMPOUNDS the **Auto rename tags** check box enable to rename the tags belonging to the object, when the tag name is built on basis of the object name.

3.13 SELECTING OBJECTS

You can select objects either from the Browse Tree or from the PFDs viewing and editing area.

Selected objects are highlighted in yellow in the PFDs.

When a PFD is open in Edit mode you can also select several objects (PFDs, UNITS, STREAMS, RESULT BOXES) by drawing a rectangle around them. Once the objects are selected you can move them together or delete them all with the **DEL** key (a confirmation of the delete will always be asked).

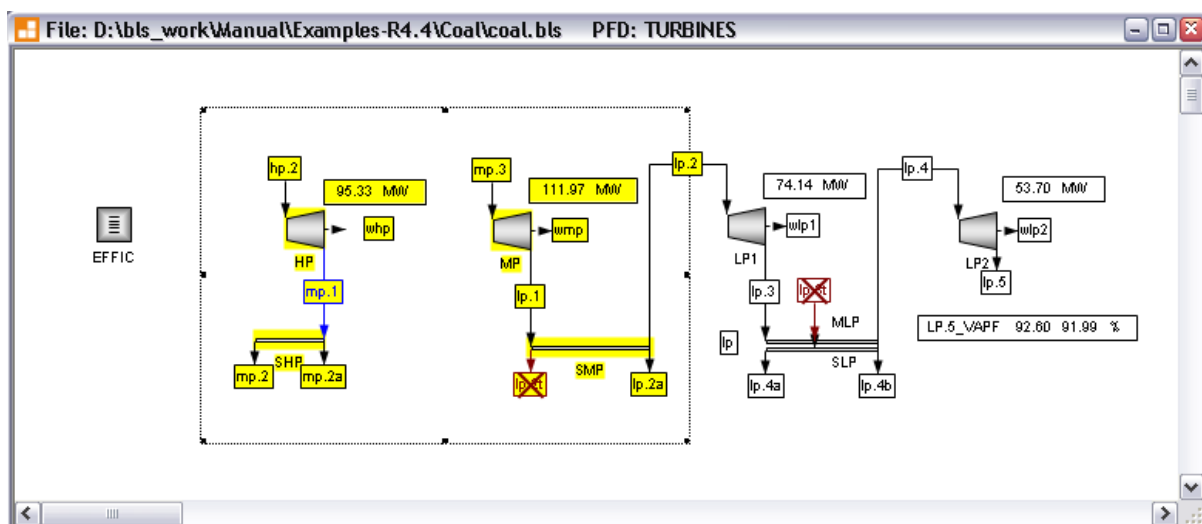


FIGURE 11 - SELECTING OBJECTS BY DRAWING A RECTANGLE

In an opened PFD you can also select several objects by clicking them one by one while maintaining the CTRL key pressed. In that case you can move the objects only with the arrows keys not with the mouse.

SECTION 4: THE MODEL FILES

Models developed with ValiModeller are saved in model files with the extension ".bls". A file can contain one or several models and some parts of a given model can be in the different BLS files.

4.1 CONTENTS OF A MODEL FILE

All the information relative to a model is stored in the model file:

- PFDS
- UNITS
- STREAMS
- THERMODS
- COMPOUNDS
- REACTIONS
- TAGS
- RESULT BOXES
- HISTORY
- User defined PHYSICAL UNITS

TAGS can also be defined through one or several TAG files (see [Using TAG Files](#)). In such a case the TAGS defined in the model file will be ignored.

The data associated to the UNITS and STREAMS correspond to the last run of ValiEngine, except if you modify them with ValiModeller. It is also possible to change UNITS and STREAMS values through measurement files. These data are used as starting points for the next run.

An empty model file called Empty.bls is delivered as a default file for the creation of your models. It contains the compounds and thermodynamic models needed for the modelling of a power plant. The file Refining.bls contains thermodynamic models relevant for refining applications.

4.2 CREATING A NEW MODEL FILE

To create a new (completely) empty MODEL FILE, use the File|New command or the  button. This will drive you through the following steps:

- define a name for the model (like with File|Save As)
- select a default plant name for the model
- load COMPOUNDS to be used in the model
- define a THERMOD (THERMODynamic model) on basis of the selected compounds

A first PFD (named PFD1) will automatically be created in the main PFD.

If you want to reuse parts of already existing models, you can also:

- open (File|Open) any exiting file you want to use as basis of your new model
- save the file (File|Save As) with a new name

It is also possible to import existing models (or part of models) as described below.

Attention: Several example files are distributed with VALI (see the examples directory on the distribution CD). They contain typical COMPOUNDS and THERMODS.

4.3 IMPORTING MODELS FROM AN OTHER MODEL FILE

To import models or part of models into another model file, you must open both files at the same time and activate the Browse Tree.

Select the PFD or the UNIT you want to copy from the source file and drag it into one of the PFDs of the target file.

You can use the same approach to copy THERMODS, REACTIONS and COMPOUNDS between two models. These objects will only be copied if their data and name are different. For example, copying the compound H₂O into a model which already refers to this compound will not create a second copy of H₂O.

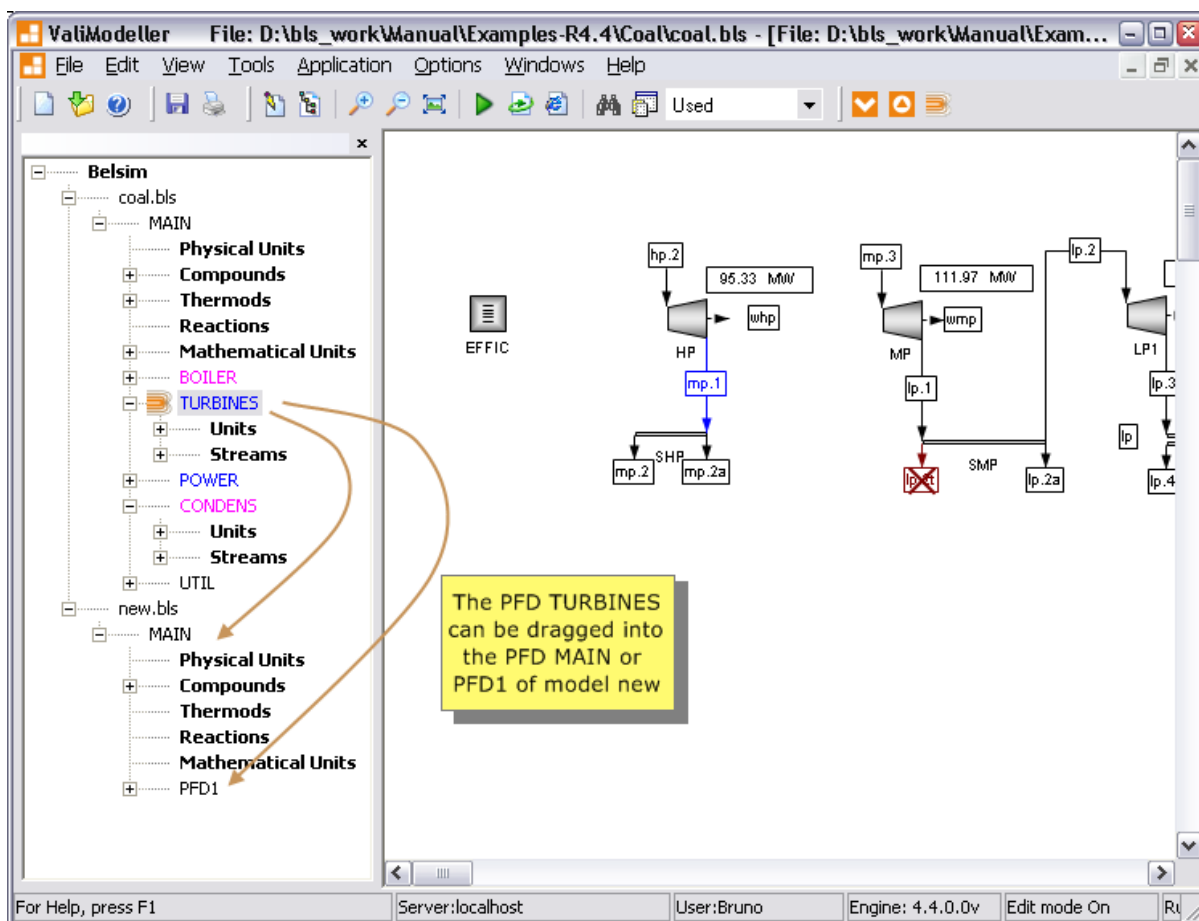


FIGURE 12 - IMPORTING A PFD

The **Rename Conflicting Objects** window is then used to resolve naming conflicts between imported and existing objects. The objects are listed by class and the **Duplicates** attribute is shown in front of the classes for which conflicts exist. You can list the conflicting objects by checking the corresponding radio button.

Conflicts must be resolved by renaming the conflicting objects. This renaming can be done for several objects at a time by entering only the first letters of the names in the field **Mask** and entering a new prefix for these objects in the field **New Name**.

The renaming can also be totally automatic by checking the **Auto rename** option. In such case the conflicting objects will be renamed according to the default prefixes. If you uncheck in addition the **Rename duplicates** only option, then all objects will be renamed, not only the conflicting ones.

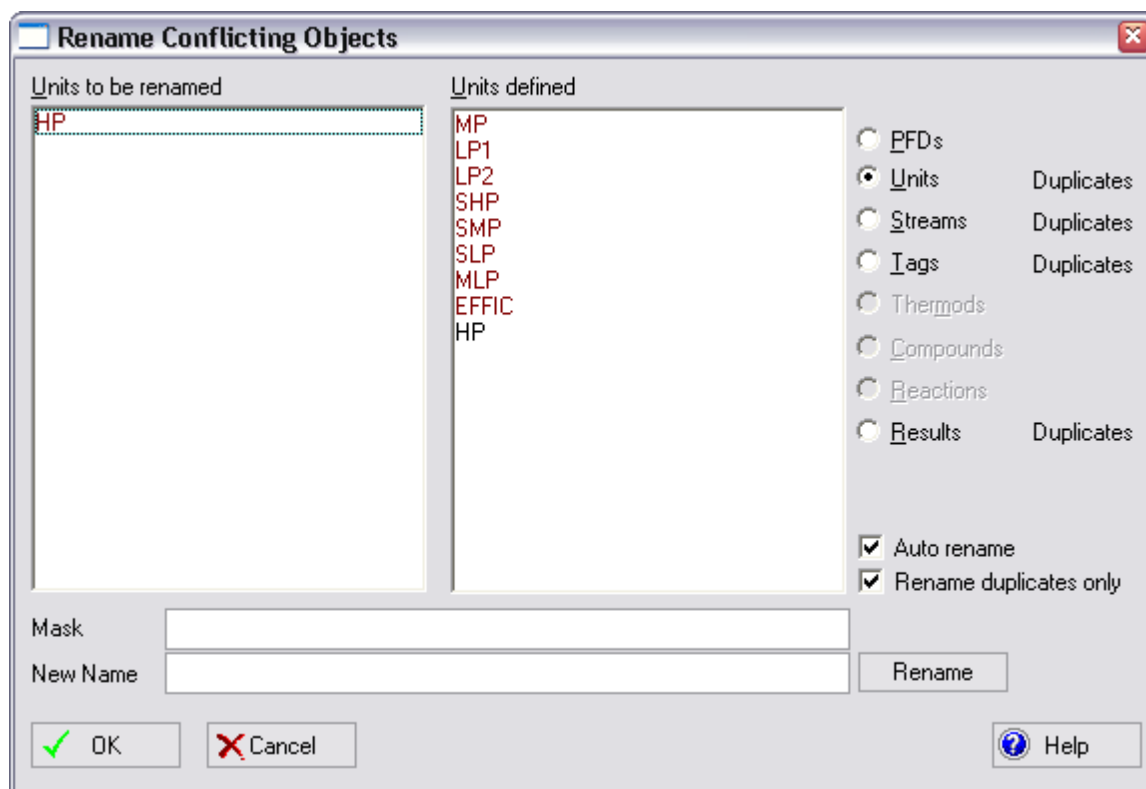


FIGURE 13 - THE RENAME CONFLICTING OBJECTS WINDOW

Attention:

1. to import several PFDs in one shot, move first all the PFDs you want to import into a single PFD and import the latter.
2. the MAIN PFD can not be imported into another model.
3. in place of importing a model, you can also define a link from an empty PFD of a model to a PFD defined in another model file. At execution time the corresponding model is then included in the global model. (see paragraph Using external PFDs in the following section)

4.4 MODEL VERSIONING AND DOCUMENTATION

The model versioning and documentation is designed for keeping track of changes made to a model. When saving the model after any change has been made to it, the Project History window is displayed. The history window can also be displayed through the **Options | History** menu command.

The model is identified by a version number and a run number.

- the version has two parts the major and the minor numbers (ex : 3.1), which are increased at user request. When the major version is incremented, the minor version number is set to 1
- the run number is increased after each successful run made with the model (provided the results are saved). This run number is reset at each new major and minor version

In addition, you can add comments to document the changes. You can also merge or delete the comments associated with past versions

Attention: it is possible to prohibit the appearance of the history window using the corresponding check box or through the user settings in the Options | Configure Notebook.

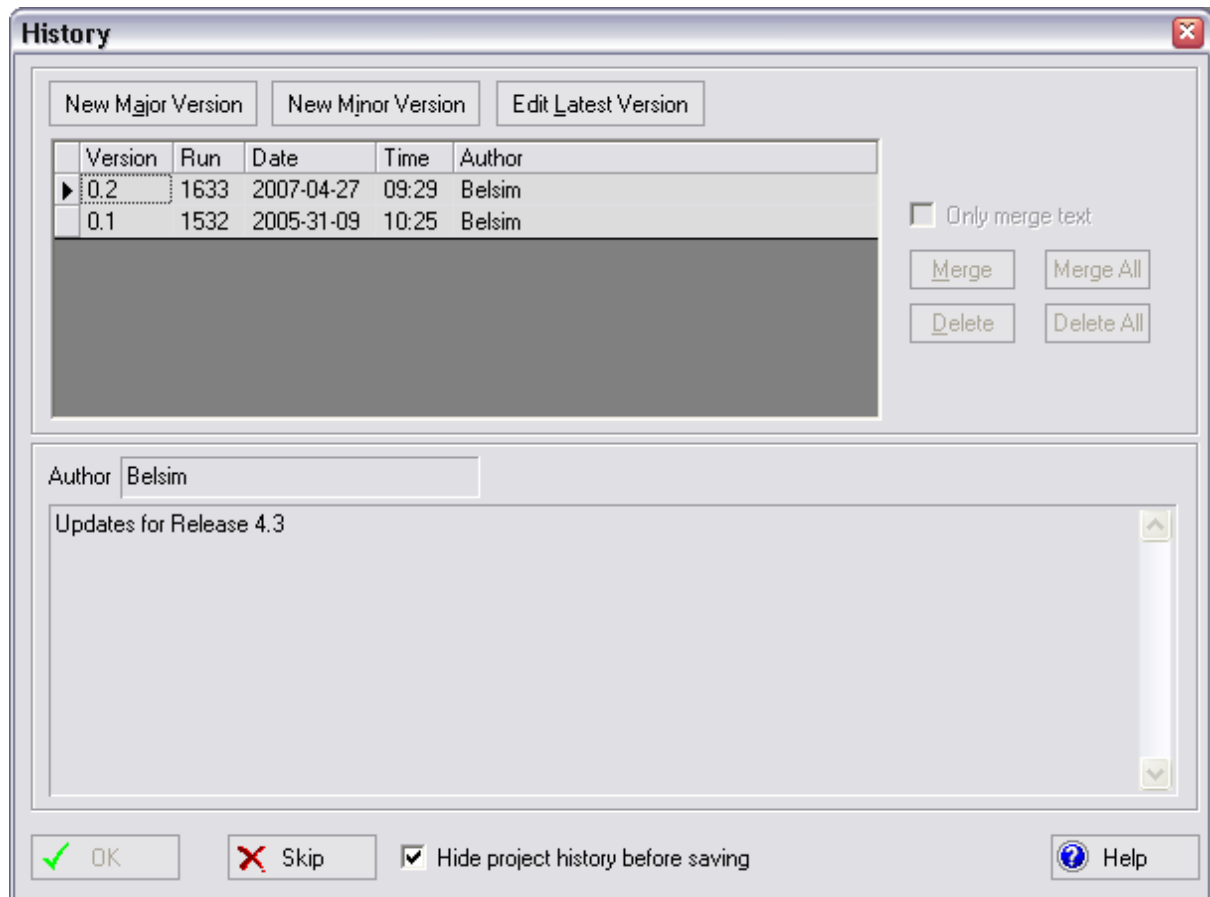


FIGURE 14 - THE PROJECT HISTORY WINDOW

4.5 THE DEFAULT PLANT

Each tag of a model must be associated to a **Plant**. This enables to easily track the tags at the License Server level. The notion of Plant is also used by ValiLink to communicate with external systems.

The default **plant** is used by default when creating new tags.

To change the **default plant**, use the **Tools | Plants** command. The **Add** enables to create new plants.

The other properties of the plants must be set though the LicenseManager.

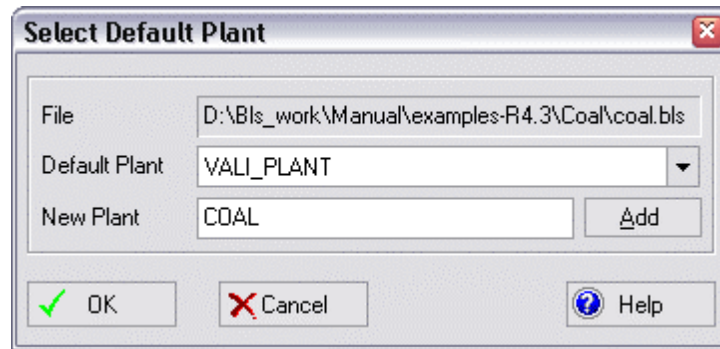


FIGURE 15 - THE SELECT DEFAULT PLANT WINDOW

SECTION 5: THE PROCESS FLOW DIAGRAMS (PFD)

Each model file consists of a number of Process Flow Diagrams (PFD).

Those PFDs are organized recursively, in a way similar to directories under Windows operating system: any PFD can contain other PFDs.

The main PFD, called by default MAIN, can contain other PFDs, but no UNITS (process equipment). The main PFD also contains special objects which allows you to change the report lay-out or mathematical options of the model like the maximum number of iterations (OPTVAL), the precision of the thermodynamic calculations (PRECIS), the default bounds on variables (BNDVAL) or the settings of the Gross Error elimination system (GROSSERR).

The other PFDs can contain:

- PFDs
- UNITS which describe the pieces of equipment of the process
- STREAMS which connect the UNITS

The PFDs are listed in the Browse Tree window. The current PFD is displayed in the View Screen. To modify a PFD you need to be in **Edit Mode** and to **Open** the PFD.

For consistency reasons, any UNIT and any PFD can appear in only one PFD. In the same way, any STREAM can connect a maximum of two different UNITS. However these two UNITS can be located in two different PFDs.

Attention: ValiModeller can be used to build models for ValiEngine but also for different applications like SIMU (sequential process simulator). However different types of PFDs are then used (SIMU PFDs in place of VALI PFDs).

5.1 CREATING A NEW VALI PFD

To create a new PFD:

- open the PFD where this new PFD must be created (you must be in Edit Mode)
- click the right mouse button anywhere in the PFD
- select the Add VALI PFD command
- point to the place where the PFD must be created and click the mouse

Alternatively you can use the Menu command **Edit|Add PFD**.

PFDs are automatically named in a sequential way : PFD1, PFD2, etc. You can however rename them as you wish. You can also change the automatic prefix in the **Options|Configure** window.

Attention: PFD names are maximum 32 characters long and are case insensitive. They must be unique with respect to other PFDs but also with respect to UNITS.

5.2 OPENING A PFD

To open a PFD:

- double-click the PFD symbol in its opened parent PFD
- click or double-click the PFD name in the Browse tree
- right-click the PFD symbol in its parent PFD and select the Open command from the Object menu
- right-click the PFD name in the tree and select the Open command from the Object menu
- click the PFD button in a UNIT or STREAM notebook
- use the Show Parent PFD from a UNIT or PFD Object Menu

The **Windows** command shows the list of opened PFDs.

Several PFDs can be opened at the same time. They are handled following the MDI (Multiple Document Interface) concept of Windows. You can switch between the opened PFDs using the **CTRL-Tab** key.

5.3 RENAMING A PFD

To rename a PFD:

- open the PFD
- right-click in any empty region of the PFD to activate the PFD object menu
- select the Rename command
- enter the new name of the PFD

You can also:

- right click the PFD to be renamed (in the PFD view or in the Browse tree)
- select the Rename command

5.4 DELETING A PFD

To delete a PFD:

- open the PFD where this PFD is referenced
- select the PFD to be deleted by clicking the mouse (or dragging a selection window)
- press the **Delete** key

You can also:

- right click the PFD to be deleted (in the PFD view or in the Browse tree)
- right click the PFD to be deleted
- select the **Delete** command

Attention: All the contents of the PFD will be deleted at the same time.

5.5 MOVING A PFD TO ANOTHER PFD

To move a PFD:

- open the PFD where this PFD is referenced
- right click the PFD
- select the **Move to Other PFD** command
- Click in another PFD to move the UNIT into this PFD

5.6 CHANGING THE STATUS OF A PFD (ON, OFF, OUT)

To change the status of a PFD:

- right click on the PFD symbol (or name in the Browse tree and select the **DATA** command
- this opens a Dialog window where the unit status can be changed

Setting a PFD OFF sets all the units and streams of the PFD (and sub-PFDs) OFF. The whole section is considered as shutdown (no flows going through the entire section).

Setting a PFD OUT sets all the units of the PFD (and sub-PFDs) OUT. The whole section is considered as not existing at all.

The order of units is also shown in this Dialog window. This order is not relevant for ValiEngine at calculation time but will affect the order of TABLES in the user defined reports (see section "[Using Measurement Files](#)").

The ZONE field is used when generating Product Balances by zones. If left blank the PFD will inherit its zone from its parent PFD. The keyword EMPTY specifies that the PFD is not included in any zone and is thus considered as being outside the overall balancing envelope.

Attention: The main PFD of a model does not have a notion of zone, so that the PFDs appearing in that main PFD will be part of the EMPTY zone, except specified otherwise through the zone field.

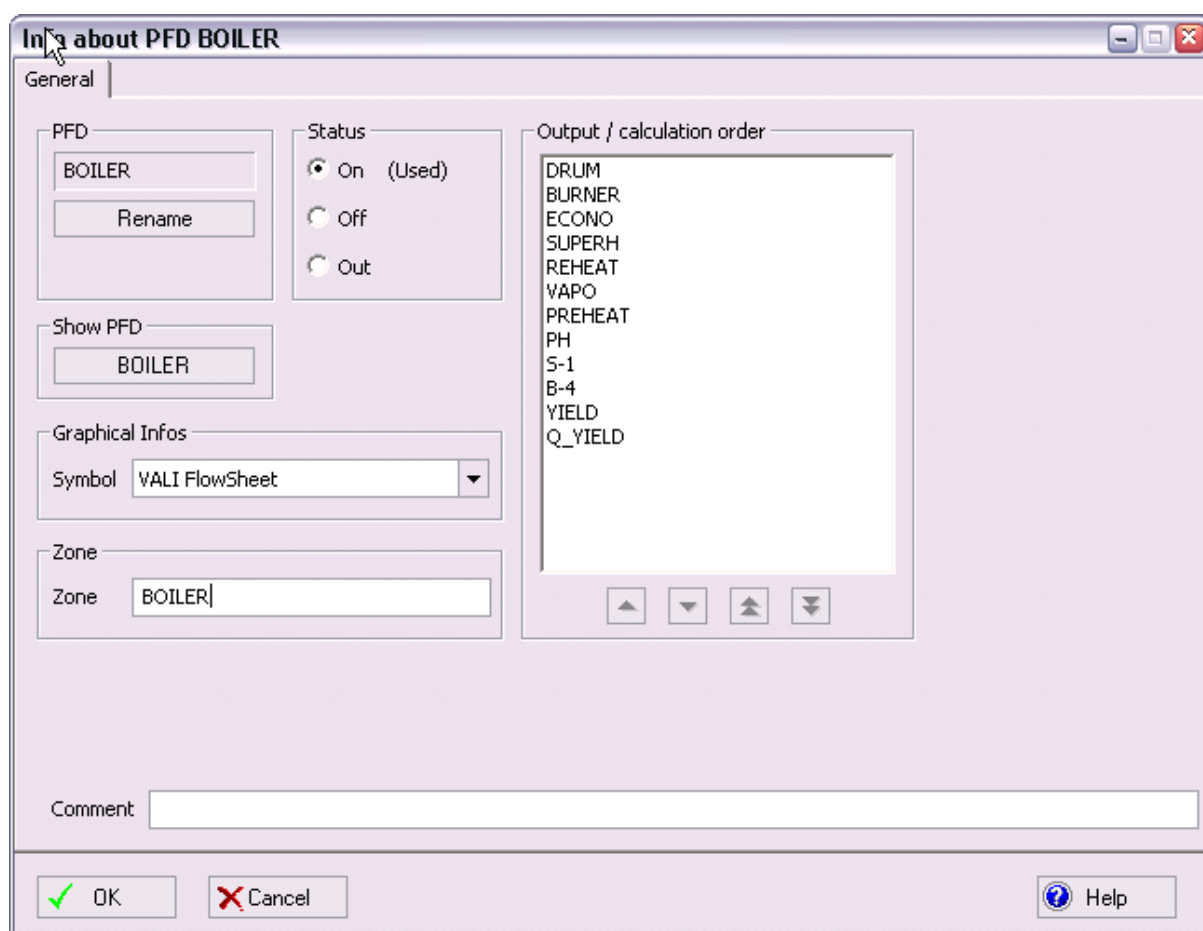


FIGURE 16 - THE PFD NOTEBOOK

5.7 USING EXTERNAL PFDs

When a PFD is empty, its DATA dialog box enables linking this PFD to a PFD defined in another model file.

When using external files, you should pay attention to the following points:

- the external model files will not be saved after a successful ValiEngine run
- objects having the same name as objects that are already defined are ignored:
 - duplicate UNITS, REACTIONS, COMPOUNDS and TAGS will be discarded
 - a STREAM can appear in two different models provided they have only one origin and only one destination unit. In addition the STREAM must use a compatible THERMOD in both file
 - if a THERMOD with the same name appears at different places, the first occurrence will be used. The duplicate thermod(s) must however be compatible with this first occurrence: referring to the same compounds in the same order

5.8 ZOOMING A PFD

Use the zooming buttons    to zoom in, out or adjust the viewing area so that the whole PFD fits into it.

5.9 PRINTING A PFD

Use the command **File | Print** to print the current PFD. Additional information like the PFD name or the date and time will be printed, according to the settings of the Configuration Notebook.

You can preview the print **File | Print Preview** or print to a file with **File | Print to File...**

You can also print all PFDS in one shot (**File | Print all PFDs**).

SECTION 6: THE UNITS

The concept of UNIT covers several aspects:

- the **UNIT TYPE** defines the way an equipment is modelled by ValiEngine : EEXVAL and HTXVAL are for example two types of UNITS allowing to model heat exchangers. The UNIT TYPE is selected at creation time; it cannot be changed afterwards, except by deleting the UNIT and creating a new one
- the **UNIT SYMBOL** is the way the UNIT is shown in a PFD. According to the type of UNIT, one or several symbols can be used. The choice of a UNIT symbol also defines the number and type of connections that the UNIT can have. For example, several heat exchangers symbols can be used for UNITS of type EEXVAL, with a various number of connected STREAMS (up to ten inlets and outlets)
- the **HOT-SPOTS** of a symbol are its possible connections and are identified by a little square that shows up when creating or reconnecting STREAMS. The **HOT-SPOTS** also defines the type of connection that can be created: inlet, outlet or both; and type of STREAM : material, mechanical or thermal
- the symbol used for a UNIT can be changed at any time (when in **EDIT Mode On**). However, you can only select between the symbols that are compatible with the already existing connections

6.1 UNIT STATUS : ON/OFF/OUT

Each UNIT has a **STATUS** that can be set (from ValiModeller or from MEA files) as:

- **ON**: the unit is operated normally (default status)
- **OFF**: the UNIT does exist but is currently shutdown so that nothing flows to or from it. As a consequence, the streams connected to that unit will be shutdown as well. In turn the fact that some streams are OFF can propagate to other units, and so forth
- **OUT**: The UNIT is considered as not existing at all. STREAMS connected only to units that are OUT becomes OUT themselves, which means that they will not be part of the model during the next run

6.2 UNIT NOTEBOOK

The **UNIT Notebook** has several pages: **General**, **Parameters**, **Tags** and **Flex code**. Depending on the UNIT type, some other pages may be accessed: **Control Tags**, **REACTIONS**, **Others**, **Composition** and **Strapping**.

The **General** page displays the UNIT name, type and comment, its status (On, Off or Out) for the next and for the last run, as well as its graphical information: symbol, position, size and orientation.

On the right side a number of drop-down boxes define various UNITS settings like the type of equations that will be generated (parameter MODE).

Clicking on the PFD button opens the PFD containing the UNIT.

Name: VAPO / BOILER **Category: Heat exchanger** **Type: HEXVAL**

General | Parameters | Tags... | Reactions | Control Tags | Others | Flex code

Unit
 VAPO
 Rename
 Copy From

Status
☒ On (Used)
☐ Off
☐ Out

PFD
 BOILER

Graphical Infos
 Symbol: Boiler element Left/Left
 Position: Horizontal 166, Vertical 356
 Size: Horizontal 22, Vertical 12
 Mirror: Horizontal ☐, Vertical ☒
 Name: Horizontal 166, Vertical 377
☒ Default
☐ Linked
☐ Default

Mode	Current Setting
Equations	Radiative exchanger
Connection rule	Cold in Tubes
Mode for U calculation	Only Uglob is variable
Number of Passes mode	1/1 or 2/2 passes
Calculation Method	DTLN
NTU Operation Mode	No Phase Change

Comment

OK Cancel Help

FIGURE 17 - THE GENERAL PAGE OF A UNIT NOTEBOOK

The **Parameters** page allows you to edit the operating conditions of the UNIT like the load or heat transfer coefficient of a heat exchanger.

On the lower part of the page, two tables show the incoming and outgoing streams. Some parameters that are linked to the inlet or outlet streams are sometimes displayed in these tables, like for example the pressure drop setting for the outlet streams. Clicking on the stream names opens their notebook.

The UNIT parameters have various roles. They can be:

- Input: this parameter is defined by the user
- Measurable: a measurement can be defined on this parameter
- Measured: a measurement is defined on this parameter
- Result: the value is calculated by ValiEngine

Attention: Some unit parameters also appear as drop down boxes in the General page.(MODE: type of equations) or in the table of the REACTIONS or Others page.

Name: VAPO / BOILER Category: Heat exchanger Type: HEXVAL

General Parameters Tags... Reactions Control Tags Others Flex code

Parameters

Name	Description	Value	Physical Unit	Status
LOAD	Heat load	232692	kW	Measured
AREA	Exchanger area	268	m2	Measured
UGLOB	Heat Transfer coeff.	0	kW/m2/K	Measurable
DTSIDE1	DT on each side	10	K	Measurable
DTSIDE2	DT on each side	10	K	Measurable

Default Values Parameters

Incoming streams

Inlet	MODEDP	DP	DPX	MASSFx	CFDP
FUMES.1	DP variable	0	-1	-1	1
VAP.1	DP variable	0	-1	-1	1

Outgoing streams

Outlet	VFRAC	U	REFERU	MASSU
FUMES.2	-100	0	0	-1
VAP.2	100	0	0	-1

OK Cancel Help

FIGURE 18 - THE PARAMETERS PAGE OF A UNIT NOTEBOOK

The **Tags** page is used to display and update the values of the measurements associated with the unit.

When in **Edit mode**, the **New**, **Copy**, **Move**, **Edit**, **Rename** and **Delete** buttons allow modifying the definitions of the tags (See section "[The tags](#)" below for more information on the handling of tags).

The **Details** button toggles the display of the tags details in the bottom part of the screen freeing more room to display additional tags in the main table.

The **Mask** combo-box allows displaying only tags where filters, compensations or bounds are defined.

The whole tag line will be highlighted in color if the measurement is either significantly corrected, suspicious, has been filtered or has been eliminated by the Gross Error Elimination procedure (If the color highlighting is activated).

The Tags page is divided into seven sub-pages:

- Measurement: values associated to the tags. Use the "Data" combo box to switch between Default, Measured and Used values
- Definition: plant name, Database name and definition of measurement: related object and corresponding parameter
- Bounds: bounds defined on the tag (SL, LL, L, H, HH, SH)
- Filters: filtering settings
- Compens: compensation settings
- KPI: used to define a tag as a KPI
- Historian: settings for the import and export to historians and to ValiDatabase

Attention: You can click on the fields of the table header to change the sorting order of the list of tags.

Name: VAPO / BOILER Category: Heat exchanger Type: HEXVAL

General Parameters Tags... Reactions Control Tags Others Flex code

Measurements Definition Bounds Filters Compens KPI Historian

Mask: None Data: Used Number of tags displayed: 5/5

Tagname	Phys. Unit	Used	Accuracy	Type	Reconciled	Accuracy	Penalty	Gain (%)	B	F	C
▶ VAPO_AREA	m2	268	CST	A	268	CST	0	0	☒	☐	☐
VAPO_DP1	bar	0	CST	A	0	CST	0	0	☒	☐	☐
VAPO_DP2	bar	0	CST	A	0	CST	0	0	☒	☐	☐
VAPO_G		1	5	%	1.02116	1.96233	0.179047	59.9231	☒	☐	☐
VAPO_LOAD	MW	0	OFF	%	232.692	1.03027	0	0	☒	☐	☐

Unit: VAPO Physical Units: AREA Plant: VALI_PLANT

Defaults: Physical Units: m2 Measured: 268 Accuracy: CST Type: % Failure Probability: 5 %

Bounds SL: 0 LL: 0 L: 0 H: 1e+010 HH: 1e+020 SH: 1e+030

Comment:

New Copy Move Edit Rename Delete Details <<

OK Cancel Help

FIGURE 19 - THE TAGS | MEASUREMENTS PAGE OF A UNIT NOTEBOOK

The **REACTIONS** page is used for the reactors of type RCTVAL, RLVVAL, RKPVAL and RKIVAL.

You modify the list of reactions by transferring them between the "Used" list and the "Defined" list

through the use of the     buttons. Multiple select is supported.

Use the     buttons to reorder the reactions.

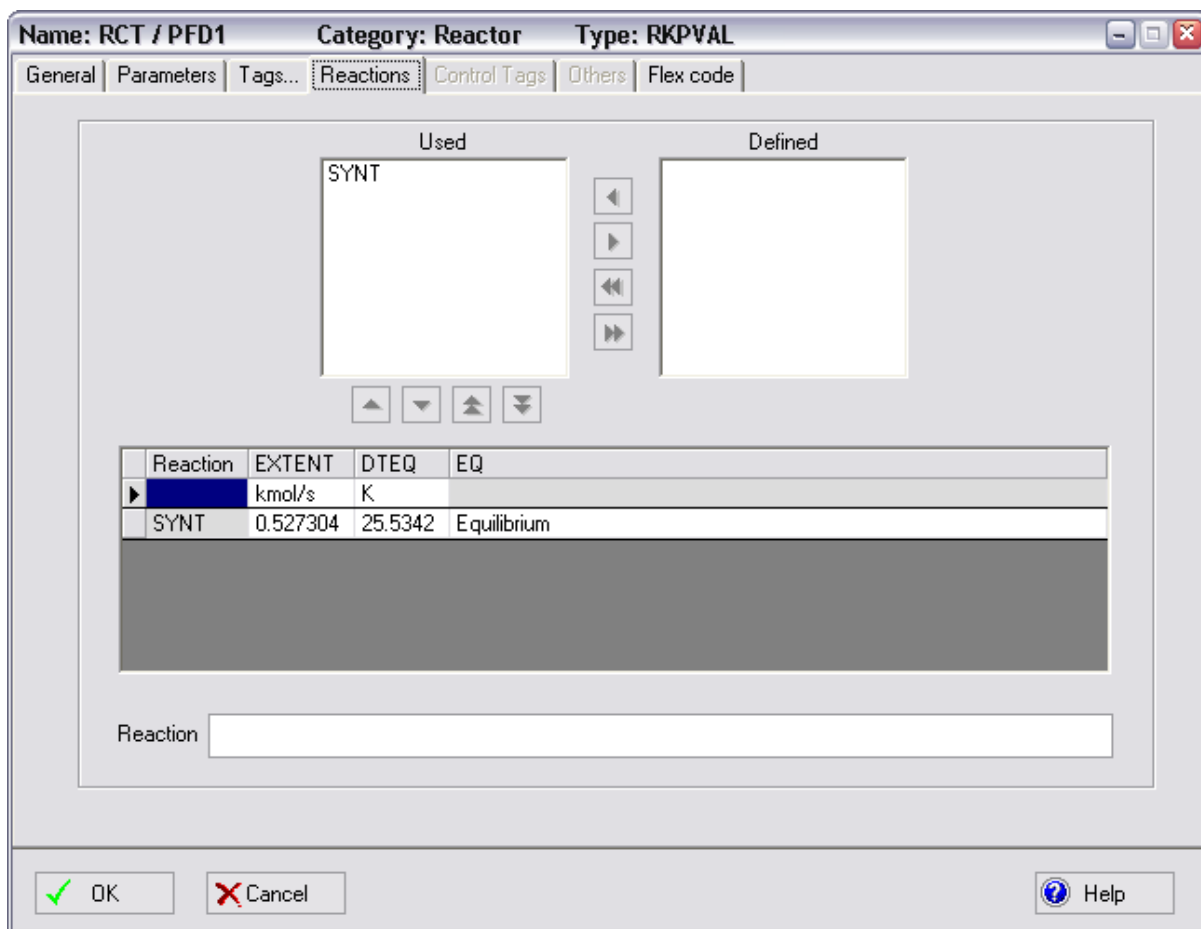


FIGURE 20 - THE REACTIONS PAGE OF A UNIT NOTEBOOK

The Others page is used to display and update the values of some more parameters, like the characteristics of the three different regions of a steam condensing preheater.

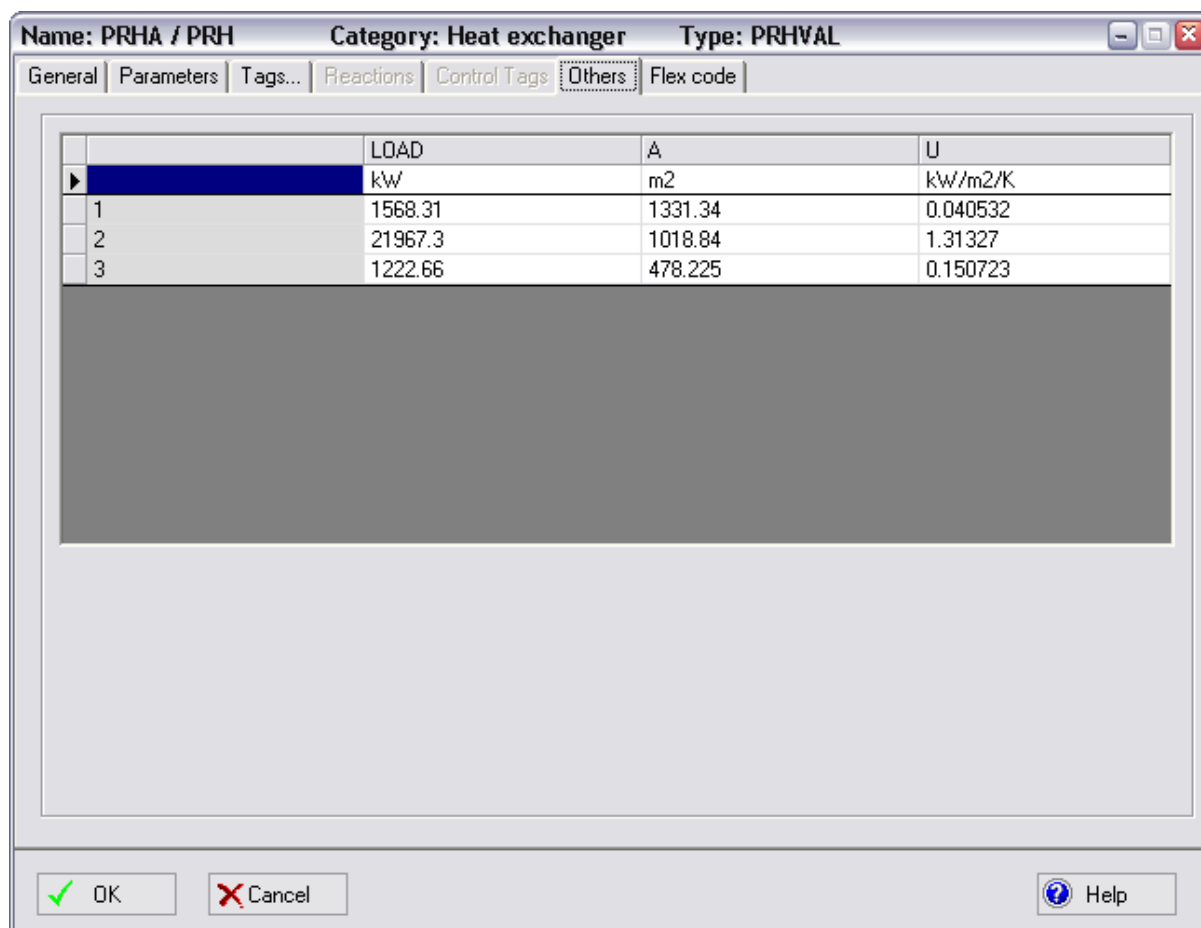


FIGURE 21 - THE OTHERS PAGE OF A UNIT NOTEBOOK

The **Flex code** page is used to display and update additional user equations defined in Flex code.

On units of type FLXVAL the calculated value is stored in one of the parameters of the unit, which can then be "measured".

On other units, the calculated value will be set to zero, defining an additional constraint.

A detailed description of the Flex code can be found in chapter "[Flex language description](#)". See also the description of unit [FLXVAL](#).

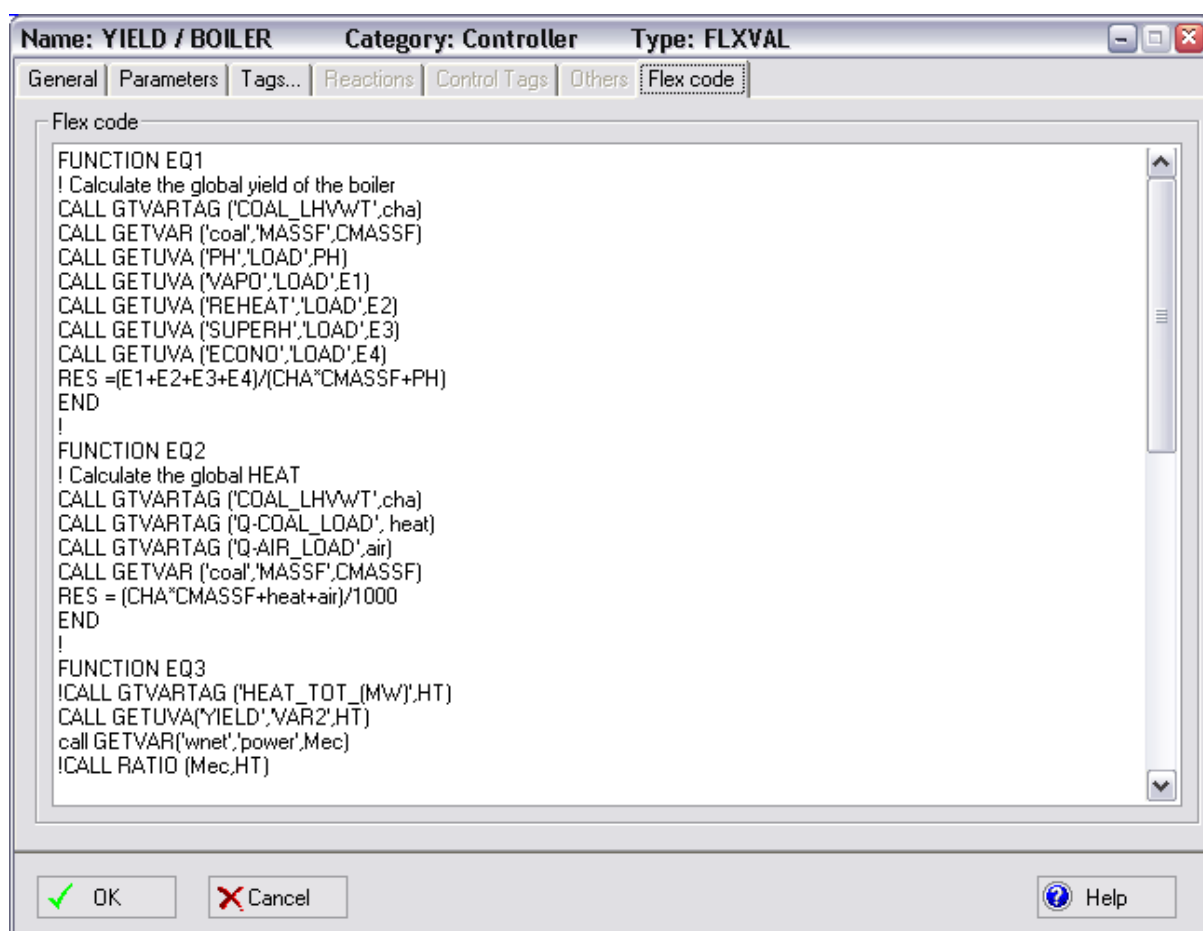


FIGURE 22 - THE FLEX CODE PAGE OF A UNIT NOTEBOOK

The **Composition** page is used only for storage tanks with variable composition.

The table displays the composition (on weight basis) at the beginning and at the end of the validation time window.

Name: TK-1 / PFD1 Category: Storage Type: TNKVAL

General | Parameters | Tags... | Reactions | Control Tags | Others | Flex code | Composition | Strapping

Compounds	Mass Initial	Mass%	Mass Final	Mass%	Delta
	kg		kg		kg
H2O	43544.100000	30.9561	245050.000000	30.1236	201505.900000
HNO3	97120.100000	69.0440	568433.000000	69.8764	471312.900000

Total 140664.000000 kg 813483.000000 kg 672819.000000 kg

OK Cancel Help

FIGURE 23 - THE COMPOSITION PAGE OF A UNIT NOTEBOOK

The **Strapping** page is used only for storage tanks.

The strapping table enables to convert tank dips (expressed in level or level %) to inventories (on volume or mass basis).

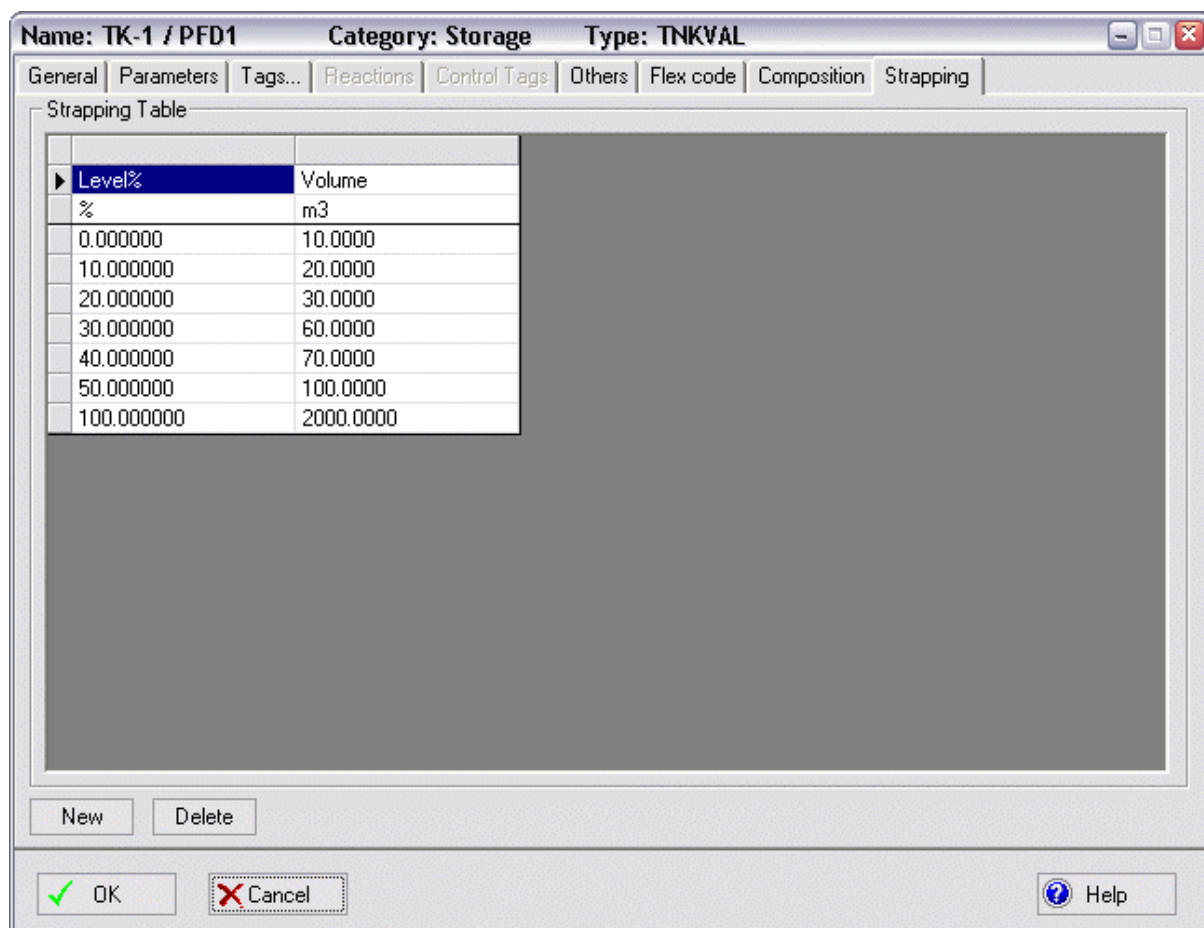


FIGURE 24 - THE STRAPPING PAGE OF A UNIT NOTEBOOK

6.3 CREATING A UNIT

To create a UNIT, you must be in **Edit mode On (Options|Edit mode)** and have opened the PFD where the UNIT must be created.

- select the **Edit|Add UNIT** menu command, or right-click in the PFD and select Add UNIT
- click at the place where you want to create the UNIT
- in the **Define New UNIT** dialog box, select the category and type of equipment and optionally update the name of the unit

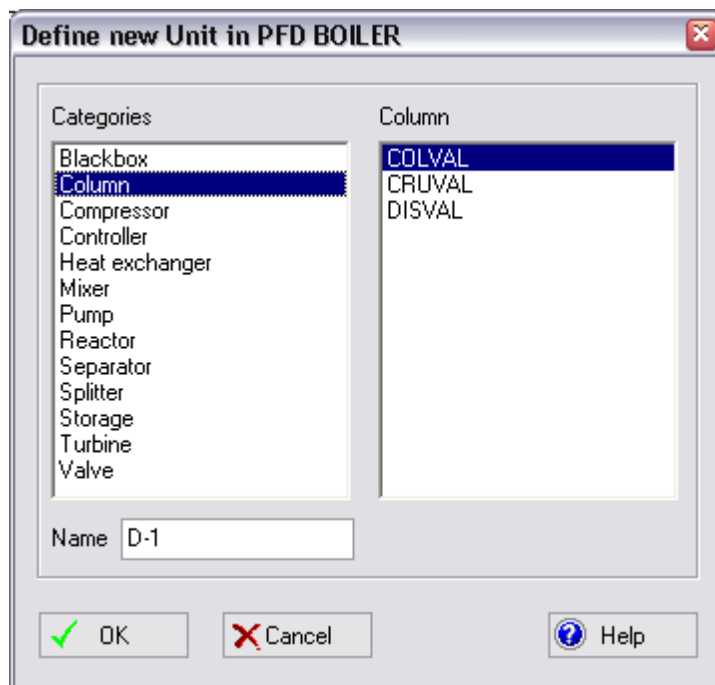


FIGURE 25 - THE DEFINE NEW UNIT DIALOG BOX

- press the **OK** button
- the UNIT NOTEBOOK is then activated
- select the appropriate symbol and its orientation
- you can optionally review the parameters of the UNIT or define reactions and tags if relevant
- click the **OK** button to confirm the creation of the UNIT or the **CANCEL** button to abort the operation

Attention:

1. Pressing the Escape key while creating a UNIT aborts the operation.
2. You can copy some (or all) settings from another unit of the same type via the Copy From button in the unit Notebook.
3. If you have activated the Add Multiple Objects option, then the Define new UNIT dialog box will open again.

You can also create a UNIT by replicating an existing unit.

- right click the existing unit
- select the Copy command
- click at the place where you want to create the UNIT (in the same or in another PFD)
- a unit is automatically created with a default name (based on the prefix)

The measurements on the unit will be copied as well.

6.4 MOVING AND SIZING UNITS

You **move** a UNIT by:

- selecting the UNIT and dragging its symbol (with the mouse or the arrow keys). You can select several objects in one shot by including them in a rectangle
- changing its position in the **General** page of its Notebook
- right-clicking the UNIT Symbol and selecting the **Move to other PFD** command. Click in another PFD to move the UNIT into this PFD

You **re-size** UNITS symbols by:

- selecting the UNIT, clicking on the edge of the button and dragging the border
- changing the X and Y size in the Notebook. The default check box allows you to restore the default size of the symbol

You can **move the UNIT name** as follows:

- select the unit name by clicking on it (or by selecting the **Select Name** command in the unit right-click menu)
- drag the unit name with the mouse or with the arrow keys

or:

- change the X and Y name position in the Notebook. The default check box allows you to restore the default position of the name

or you can reset its default position:

- right-click the UNIT Symbol or Name and select the **Auto position Name** command

6.5 DELETING UNITS

To delete a UNIT:

- select the UNIT
- press the **Delete** key

Alternatively you can also select the **Delete** option from the Menu activated by right clicking on the unit.

Attention:

1. When deleting a UNIT, all the STREAMS that are connected only to this UNIT are also deleted.
2. Depending of your configuration settings, you will be asked to confirm the deletion.

Alternatively you can also select the **Delete** option from the Menu activated by right clicking on the unit.

6.6 RENAMING UNITS

To delete a unit:

- right-click the UNIT
- select the **Rename** option
- enter the new name
- check the auto-rename tag definitions box to automatically rename the tags associated with the unit
- press the **OK** button

Alternatively you can also use the Rename button from the UNIT Notebook.

6.7 CREATING A FLEX PROGRAM FROM AN EXTERNAL FILE

The Flex code associated to a unit is normally edited directly in the Flex code page of the unit notebook.

However, the Flex code can also be defined in an external file as follows:

- define the unit of type FLXVAL
- write your FLEX program in a separate file and encapsulate it between the following two lines:

```
&&&COMMENT <unit>
...
&&&END
```

where <unit> is the unit name

- when starting your model with ValiEngine enter the name of the file into the corresponding field

Attention:

1. When ValiEngine saves the results after a successful run, the Flex program is written into the MODEL FILE file. The program is then kept for further runs and the option -rea does no longer have to be used anymore.
2. To modify the Flex program of a unit, you must reload it or edit it from the Flex code page.

SECTION 7: THE STREAMS

There are three types of STREAMS: material, mechanical and thermal. As for UNITS, STREAM data are accessed via a **Notebook**.

Each STREAM can connect a maximum of two UNITS, going from one UNIT to the other one. Three cases are therefore possible:

- the STREAM is an inlet of one UNIT
- the STREAM is an outlet of one UNIT
- the STREAM is an outlet of one UNIT and an inlet of an other UNIT. These two UNITS can be in two different PFDs

7.1 STREAM NOTEBOOK

The Notebooks of thermal and mechanical STREAMS have a **General** and a **Tags** page. In addition, material STREAMS have a **Composition** page. The top line of the Notebook indicates the origin and destination of the STREAMS.

The **General** page presents the STREAM name and status (ON, OFF or OUT) and its parameters. For material STREAMS, the page also displays the associated THERMOD, the STREAM state and initialization mode as well as information relevant to products balancing:

- Product code to be used for the following run and used during the previous run.
- Movement type.
- Sub-Movement type.

The movement type can be any of the following:

- NONE: the stream will not be considered in any product balance.
- AUTO: the stream will be part of the product balance and considered as production, consumption etc... depending on the units it is connected to.
- TANK_CHARGE, TANK_RUNDOW, PRODUCTION, CONSUMPTION, IMPORT, EXPORT, TRANSFER : selects the corresponding movement type.

When a stream goes from Zonefrom to Zoneto, its Movement Type (unique) results in multiple actions:

ZONEFROM/TO	MVT_TYPE	ACTION
X/Y	PRODUCTION	Add to TRAN-OUT(ZoneFrom) Add to TRAN-IN (ZoneTo) Add to PRODUCTION (ZoneFrom)

X/Y	CONSUMPTION	Add to TRAN-OUT(ZoneFrom) Add to TRAN-IN (ZoneTo) Add to CONSUMPTION (ZoneTo)
X/Y	TANK_CHARGE	Add to TRAN-OUT(ZoneFrom) Add to TRAN-IN (ZoneTo) Add to TANK_CHARGE (ZoneTo)
X/Y	TANK_RUNDOWN	Add to TRAN-OUT(ZoneFrom) Add to TRAN-IN (ZoneTo) Add to TANK_RUNDOWN (ZoneFrom)
X/Y	IMPORT	Add to IMPORT (ZoneTo)
X/Y	EXPORT	Add to EXPORT (ZoneFrom)
X/Y	TRANSFER	Add to TRAN-OUT(ZoneFrom) Add to TRAN-IN (ZoneTo)

TABLE 1 - MOVEMENT TYPES

The sub-movement type is linked to a movement type. For instance, a Movement of type EXPORT, defined as a stream going out to nowhere, from a unit or a tank, could be split into 3 sub-movements, for example: SHIPMENT, LOSSES and other real EXPORT.

The sub-movement must be defined manually through ValiModeller or through a measurement file.

Clicking on the PFD button(s) with the origin and destination PFDs opens the corresponding PFDs.

Material Stream: AIR.0 From: E-2 / UTIL To: PH / BOILER

General | Composition | Tags...

Stream: AIR.0
 Rename
 Copy From

Status:
☒ On
☐ Off
☐ Out

Thermod: AIR
 State: Liquid/Vapour
 Init: Pressure and Temperature fixed

PFD:
 UTIL BOILER

Product:
 Product: UAIR Used Product:
 Movement: AUTO Sub-Movement:

Name	Description	Value	Physical Unit	Status
T	Temperature	15.0866	C	Measurable
P	Pressure	1.0128	bar	Measurable
VAPF	Vapour fraction	100	%	Measurable
H	Enthalpy	-287.463	kJ/kmol	Measurable
S	Entropy	0	kJ/kmol/C	Result
A	Exergy	-1342.42	kJ/kmol	Measurable
V	Volume	0	m3/kmol	Result
MOLW	Molecular weight	28.7957	kg/kmol	Measurable
HW	Weight enthalpy	-9.98283	kJ/kg	Measurable
MOLF	Molar Flowrate	10.1713	kmol/s	Measurable
MASSF	Mass Flowrate	305.2	kg/s	Measurable

Comment:

OK Cancel Help

FIGURE 26 - THE GENERAL PAGE OF A STREAM NOTEBOOK

For material STREAMS, the **Composition** page displays the composition in terms of partial rates and fractions, on a molar and on a weight basis.

The molar and weight fractions as well as the total rates can be modified by the user and will be used as starting point for the next run.

The total flow rates can also be edited from the **General** page.

Material Stream: AIR.0 From: E-2 / UTIL To: PH / BOILER

General Composition Tags...

Composition

Compounds	Moles kmol/s	Moles%	Mass kg/s	Mass%
O2	2.215540	20.8942	70.894618	23.2183
N2	8.334640	78.6020	233.481581	76.4663
H2O	0.053453	0.5041	0.962977	0.3154

Total 10.603600 kmol/s 305.339000 kg/s

OK Cancel Help

FIGURE 27 - THE COMPOSITION PAGE OF A STREAM NOTEBOOK

The **Tags** page is used to display and update the values of the measurements associated with the stream.

When in **Edit mode**, the **New**, **Copy**, **Move**, **Edit**, **Rename** and **Delete** buttons allow modifying the definitions of the tags.

The **Details** button toggles the display of the tags details in the bottom part of the screen freeing more room to display additional tags in the main table.

The **Mask** combo-box allows displaying only tags where filters, compensations or bounds are defined.

The whole tag line will be highlighted in color if the measurement is either significantly corrected, suspicious, has been filtered or has been eliminated by the Gross Error Elimination procedure (If the color highlighting is activated).

The Tags page is divided into seven sub-pages:

- Measurement: values associated to the tags. Use the "Data" combo box to switch between Default, Measured and Used values
- Definition: plant name, Database name and definition of measurement: related object and corresponding parameter
- Bounds: bounds defined on the tag (SL, LL, L, H, HH, SH)
- Filters: filtering settings
- Compens: compensation settings
- KPI: used to define a tag as a KPI
- Historian: settings for the import and export to historians and to ValiDatabase

Attention:

1. You can click on the fields of the table header to change the sorting order of the list of tags.
2. See section "[The Tags](#)" for more information on the handling of tags.

Material Stream: AIR.0 From: E-2 / UTIL To: PH / BOILER

General | Composition | Tags...

Measurements | Definition | Bounds | Filters | Compens | KPI | Historian

Mask: None Data: Used Number of tags displayed: 7/7

Tagname	Phys. Unit	Used	Accuracy	Type	Reconciled	Accuracy	Penalty	Gain (%)	B	F	C
AIR.0_HUMIDITY	%	30.0014	2	%	30.0073	1.99952	9.7206...	0.0040...	☐	☐	☐
AIR.0_MG02	%	21	CST	A	21	CST	0	0	☐	☐	☐
AIR.0_T	C	15.1468	0.5	A	15.0867	0.493036	0.0144...	1.39285	☒	☐	☐
AIR.0_P	bar	1.013	2	%	1.0128	2.00031	9.7271...	0.0040...	☐	☐	☐
AIR.0_MOLF	Nm3/h	867783	10	%	855209	0.657867	0.0209...	93.5167	☒	☐	☐
AIR.0_HFLOWTOT	MW	-0.6722	OFF	%	-0.696933	33.8213	0	0	☐	☐	☐
AIR.0_MASSF	kg/s	306.08	OFF	%	305.202	0.657625	0	0	☐	☐	☐

Stream: AIR.0 HUMIDITY Plant: VALI_PLANT

Physical Units: Measured Accuracy: Type: Failure Probability:

Defaults: % 30.0014 2 % 5 %

Bounds SL 0 LL 0 L 0 H 100 HH 100 SH 100

Comment:

New Copy Move Edit Rename Delete Details <<

OK Cancel Help

FIGURE 28 - THE TAGS/MEASUREMENTS PAGE OF A STREAM NOTEBOOK

7.2 CREATING A STREAM

To create a STREAM, you must be in **EDIT** mode and have already opened the PFD(s) where the STREAM must be created.

Depending on the status of the **Add multiple objects** option, you can either create the streams one by one via the **Edit | Add STREAM** menu (add single stream) or create a number of streams in one shot. In the latter case, the **Define New STREAM** dialog box re-appears after each stream creation.

- select the **Edit|Add STREAM** command

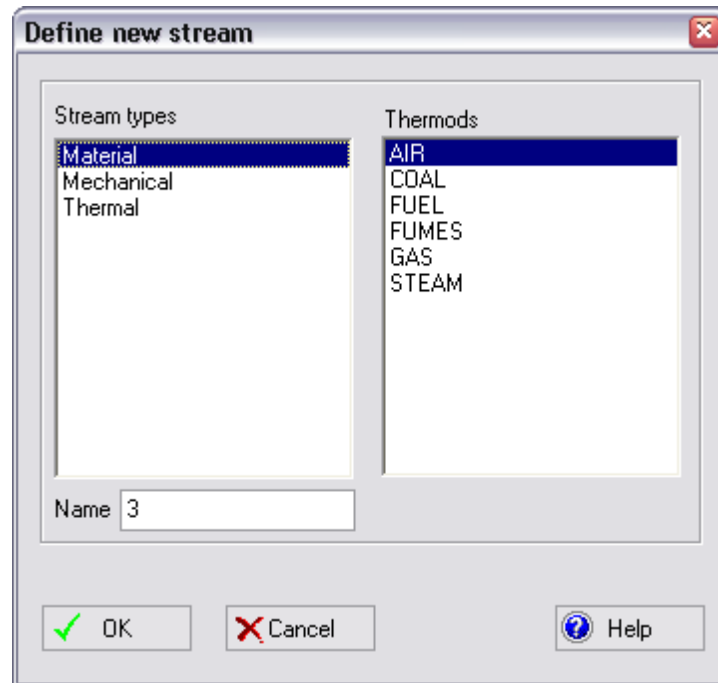


FIGURE 29 - THE DEFINE NEW STREAM DIALOG BOX

- select the Type of STREAM: material, mechanical or thermal
- if creating a material STREAM, select the THERMOD
- press the OK button
- the outlet hot-spots which are compatible with that type of STREAM are then activated in the current PFD and the mouse pointer is replaced by a cross
- enter the stream connections as indicated below:
 - for a STREAM coming out of a UNIT:
 - click on a unit outlet hot-spot.
 - click again at the same place, to terminate the stream creation or, if needed add fixed points to the stream by clicking at the required points in the PFD and terminate the stream creation by double clicking on the last point
 - for a STREAM going from one unit to another UNIT:
 - click on a hot-spot of the origin unit
 - if needed add fixed points to the stream by clicking at the required points in the PFD
 - if needed activate the PFD containing the destination unit
 - click on a hot-spot of the destination unit

- for a STREAM going into a UNIT
 - toggles destination hot-spots by pressing the space bar
 - if needed add fixed points to the stream by clicking at the required points in the PFD
 - click on a hot-spot of the destination unit
 - the STREAM NOTEBOOK is then activated
 - you can copy the stream settings from another stream, using the **Copy From** button
 - you can also review the parameters and measurements of the STREAM
 - click the **OK** button to confirm the creation of the STREAM or the **CANCEL** button to abort the operation

Attention: Pressing the Escape key while creating a STREAM aborts the operation.

A **STREAM name** box is automatically created. The STREAM name box can be moved by selecting it and dragging it (with the mouse or the arrows) to any place. Once moved, the button becomes static. This means that it does not move when moving the UNITS where the STREAM is connected. Use the **Autodraw** or the **Auto position name** command to restore the automatic STREAM Button positioning.

7.3 REDRAWING AND RECONNECTING STREAMS

Attention: Take care that reconnecting streams to units can affect their connection order. This can in turn change the meaning of unit parameters such as REFDP, DPi, VFRACi, etc.

To **redraw** a STREAM automatically, keeping its existing connections:

- right-click the STREAM box or any section of the stream line
- select the **Auto-draw** option: all points entered for the STREAM are deleted and the STREAM is redrawn

To **redraw a STREAM manually**, keeping its existing connections, or changing its final connection:

- right-click the STREAM button
- select the **Disconnect To** option
- right-click again and select the **Autodraw** option: all points entered for the STREAM are deleted
- enter new points
- terminate your input by clicking on the hot-spot where the STREAM must be connected

To **add a connection** for a STREAM:

- right-click the **STREAM** button
- select the **Connect From** or the **Connect To** option
- enter fix points (if needed)
- click on the hot-spot where the **STREAM** must be connected

To **reconnect** a **STREAM**:

- right-click the **STREAM** button
- select the **Reconnect To** or **Reconnect From** option: the corresponding connection and the points of the **STREAM** are deleted; you can then enter the new connection

To **Disconnect** a **STREAM**:

- right-click the **STREAM** button
- select the **Disconnect To** or **Disconnect From** option

7.4 DELETING STREAMS

To delete a **STREAM**:

- select the **STREAM** by clicking its box or any section of its line
- press the **Delete** key

Alternatively you can also select the **Delete** option from the Menu activated by right clicking on the stream.

7.5 RENAMING STREAMS

To rename a stream:

- right-click the **STREAM** box or any section of its line
- select the **Rename** option
- enter the new name
- check the **auto-rename tag definitions** box to automatically rename the tags associated with the stream
- press the **OK** button

7.6 NAVIGATING BETWEEN PFDs

When a stream has connections in two different PFDs, you can easily go from one PFD to another:

- right-click the **STREAM** button or any section of its line
- select the **From PFD** (or the **To PFD**) option

SECTION 8: THE TAGS

Tags can be defined either: in the model itself, using the **New**, **Copy**, **Move** and **Edit** buttons of the Tags page of the UNITS and STREAMS Notebooks or in a separate file, called a TAG file (see [Using TAG Files](#)). You must be in **Edit mode** to create or modify Tags in the model itself.

The Tags Notebook is divided into seven pages:

- Measurements: values associated to the tags. Use the "**Data**" combo box to switch between Default, Measured and Used values
- Definition: related object and corresponding parameter; failure probability, bound control settings
- Bounds: bounds defined on the tag (SL, LL, L, H, HH, SH)
- Filters: filtering settings
- Compens: compensation settings
- KPI: used to define a tag as a KPI
- Historian: plant name, DB name and other settings for the import and export to historians and to ValiDatabase

Attention:

1. See [Process Measurements](#) for more information on measurements
2. See also section **Viewing and editing Measurements Values** below

8.1 SETS OF VALUES ASSOCIATED TO A TAG

In a model, four sets of values and precisions are attached to each tag:

- the default value and its precision, which are the values as defined in the model (or in the tag file) by the user
- the measured value and its precision, which are initially set to the default values but can be overwritten by values coming from measurement files (MEA files) by ValiEngine
- the used value and its precision, which are the values actually used during the reconciliation step. They are initialized to the measured values but can be overwritten by filters, compensation algorithms or control strings
- the reconciled value and its precision as validated by ValiEngine

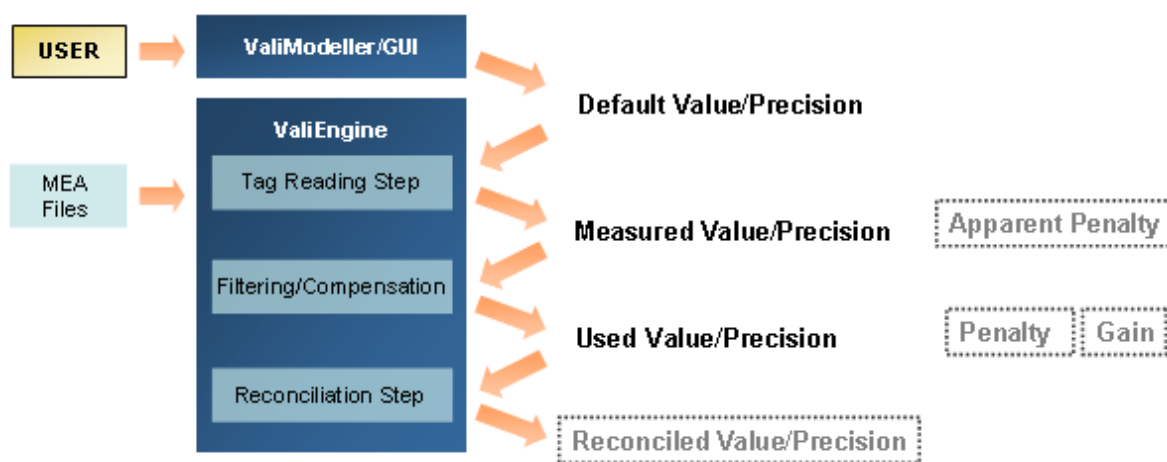


FIGURE 4 - THE TAG VALUES

In addition, ValiEngine will calculate for each tag:

- the apparent penalty: calculated on the basis of the measured value:

$$\text{ApparentPenalty} = \left(\frac{\text{reconciled}_i - \text{measured}_i}{\sigma_{i,\text{measured}}} \right)^2$$

- the penalty: calculated on the basis of the used value, can be selected (through the mathematical unit OPTVAL) as being either the measurement penalty, the correction penalty or the measurement impact:

$$\text{MeasurementPenalty} = \left(\frac{\text{reconciled}_i - \text{used}_i}{\sigma_{i,\text{used}}} \right)^2$$

$$\text{CorrectionPenalty} = \left(\frac{\text{reconciled}_i - \text{used}_i}{\sigma_{i,\text{correction}}} \right)^2$$

Impact = decrease in total penalty when setting the measurement OFF

- the gain: which expresses the accuracy improvement and is calculated on the basis of the reconciled and the used accuracies:

$$\text{Gain} = 100 \cdot \left(\frac{\sigma_{i,\text{used}} - \sigma_{i,\text{reconciled}}}{\sigma_{i,\text{used}}} \right)$$

8.2 CREATING NEW TAGS

To create a new tag, press the New key in the Tags page of the corresponding unit or stream.

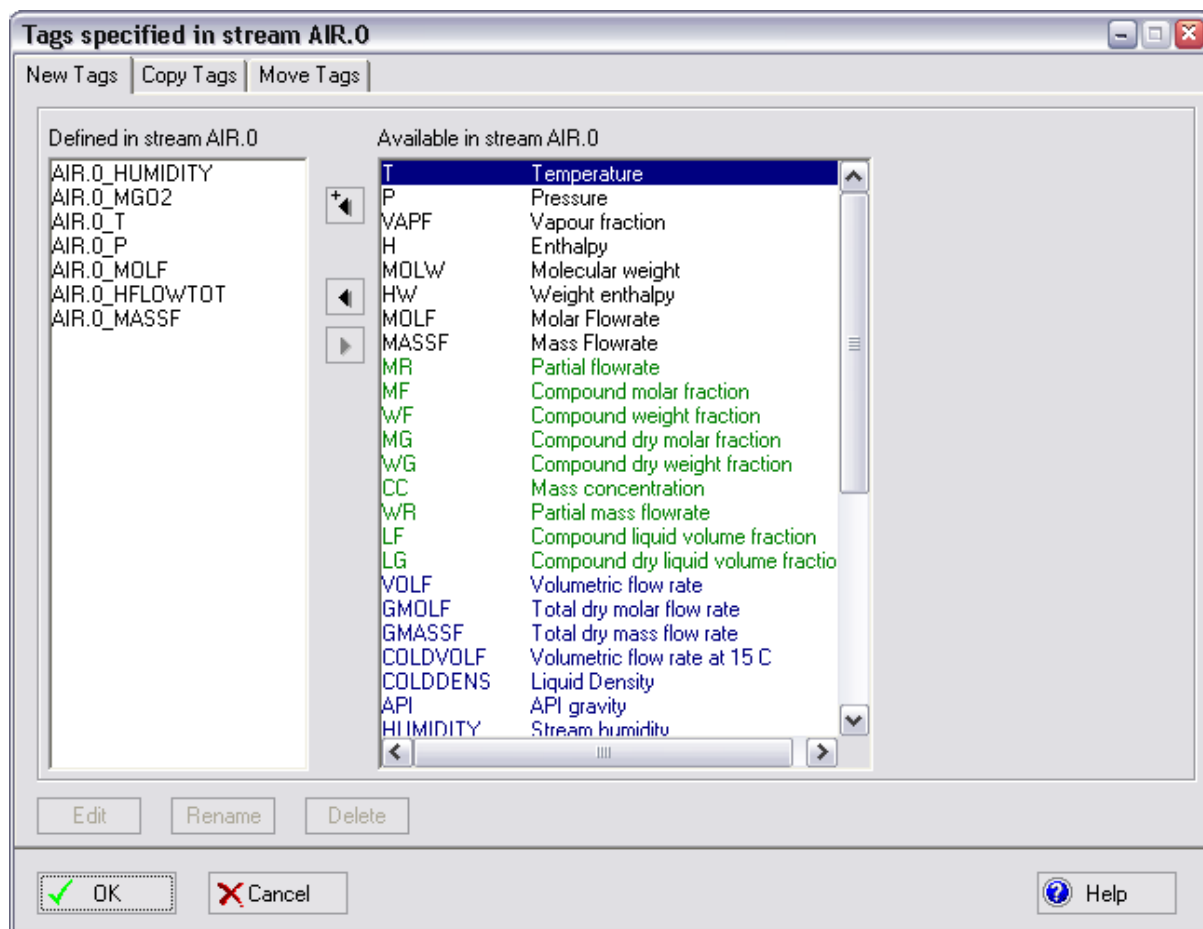


FIGURE 30 - THE NEW PAGE OF THE TAGS NOTEBOOK

To create a new tag:

- select the type of Tag in the table labeled "Available in..."
- if necessary, complete the information from the secondary table, like for example the compounds for composition measurements
- press the < button to confirm the creation of the Tag or press the +< button to confirm the creation of the Tag and to open the edition window and update values

Attention: if a Tag with the same name already exists, a new unique name is automatically generated.

You can create several tags in a row. When all tags have been defined, press the OK button to close the window.

8.3 EDITING TAGS

You access the **Edit** window by clicking on the Edit button of the Tags page of the UNITS and STREAMS Notebook or by double-clicking the tag name in the Tag Notebook.

The **Edit** window displays all the values associated to a tag:

- name of the tag
- name to be used when acquiring the tag value through ValiLink
- related plant
- definition of the tag
- physical units
- default value, accuracy and failure probability factor. The failure probability is a weighting factor for the gross error elimination procedure. Use higher factor to increase the probability that the measurement will be removed. Use a failure probability of 0 to prevent any elimination
- filter settings (see more information below)
- bounds: 3 levels are considered for both High and Low bounds:
 - warning (L -> H): a warning message will be issued if the measured or the validated values are outside this range
 - error (LL -> HH): an error message will be issued if the measured or the validated values are outside this range. (ValiFilter uses these bounds)
 - solver (SL -> SH): a fatal error occurs if the measured value is outside this range. The validated value will be kept between these bounds
- compensation settings (see more information below)
- KPI settings: all tags specified as "KPI" will be printed out in the html report with their reconciled and their target value. The settings also enable calculating the raw value and precision of the KPI, as calculated on basis of measured values
- tag comment

The Tag Comment will be printed out in the html (or r4v) reports. This comment can also be used to define control strings, to carry out various actions depending on the measured values, and sometimes validated value, of a given Tag: issuing messages, replacing the measured value by another value or changing its accuracy. (See more information in the section Using **Control Strings** below).

Material Stream: AIR.0 From: E-2 / UTIL To: PH / BOILER

Definition

Tag name:

DB name: Plant:

Tag owner: Description:

Values

Phys.Units:

Defaults: Measured: Accuracy: Type: Failure Probability: %

Filter: % CUTOFF:

Bounds: SL: LL: L: H: HH: SH:

Advanced

Compens: Compens: Bias: Aref: Bref: Cref:

KPI Mode: Target:

Comment:

FIGURE 31 - THE EDIT PAGE OF A TAG

FILTER SETTINGS

Filters are configured as follows:

Measured defines the action on the measured value and can be:

- NONE: no action is taken
- BND: the measured value is replaced by the activated bound
- DEF: the measured value is replaced by the default value
- MID: the measured value is replaced by the average of the bounds $(LL+HH)/2$
- LAST: the measured value is replaced by the validated value (as obtained during the last successful and saved run)
- Value: the measured value is replaced by the given value

Accuracy and **Type** defines the action on the accuracy and can be:

- NONE: no action is taken
- OFF: the tag is set OFF
- DEF: the accuracy is replaced by the default value
- MID: the accuracy is replaced by half the difference between the bounds (HH-LL)/2
- Value: the accuracy is replaced by the given value

Failure Probability defines the action on the Failure probability and can be:

- NONE: no action is taken
- Value: the failure probability is replaced by the given value

CutOff: if the measured value is below the specified value, the corresponding object (STREAM or UNIT) will be set OFF.

COMPENSATION SETTINGS

The compensation mechanism corrects measured values which are known to be deviating:

- because of varying operating conditions of the sensor against reference conditions
- because of an identified bias of the measured value

The compensation factor is defined as follows:

$$Y_{used} = Y_{measured} \cdot COMPENS + BIAS$$

Where:

- Y_{used} : is the compensated value
- $Y_{measured}$: is the measured value
- COMPENS: is a proportional compensation factor, **expressed in percentage**
- BIAS: is a constant bias

Depending on the compensation mode, COMPENS can be a constant user given value or can be calculated from a compensation correlation in function of operating and reference conditions (AREF, BREF and CREF).

COMPENSATION MODE	DESCRIPTION
NONE	No compensation (default)
BIAS	User given COMPENS and BIAS Can be used for any type of measurement

COMPENSATION MODE	DESCRIPTION		
	Differential Pressure Flow Meter (Orifice, Venturi, Nozzle, etc.)		
	MASSF	MOLF	COLDVOLF CONVVOLF
DP_D15 (liquid flow) AREF=ref. density at 15°C (kg/m3) BREF=ref. density at flow T,p CREF=ref. MW	$\sqrt{\frac{\rho}{B_{REF}}}$	$\sqrt{\frac{\rho}{B_{REF}} \frac{C_{REF}}{MW}}$	$\sqrt{\frac{\rho}{B_{REF}} \frac{A_{REF}}{\rho_{15}}}$
DP_D0 (gas flow) AREF=ref. density at 0°C (kg/Nm3) BREF=ref. density at flow T,p	$\sqrt{\frac{\rho}{B_{REF}}}$	$\sqrt{\frac{\rho}{B_{REF}} \frac{A_{REF}}{\rho_N}}$	Not available
DP_TPD0 (gas flow) AREF=ref. temperature (K) BREF=ref. pressure (bara) CREF=ref. density at 0°C (kg/Nm3)	$\sqrt{\frac{A_{REF} \cdot p}{T \cdot B_{REF} \cdot C_{REF}}}$	$\sqrt{\frac{A_{REF} \cdot p \cdot C_{REF}}{T \cdot B_{REF} \cdot \rho_N}}$	Not available
DP_TPMW (gas flow) AREF=ref. temperature (K) BREF=ref. pressure (bara) CREF=ref. MW (kg/kmol)	$\sqrt{\frac{A_{REF} \cdot p \cdot MW}{T \cdot B_{REF} \cdot C_{REF}}}$	$\sqrt{\frac{A_{REF} \cdot p \cdot C_{REF}}{T \cdot B_{REF} \cdot MW}}$	Not available
	Volumetric flow meter (Vortex, Turbine, etc.)		
	MASSF	MOLF	COLDVOLF CONVVOLF
VOL_D15 (liquid flow) AREF=ref. density at 15°C (kg/m3) BREF=ref. density at flow T,p CREF=ref. MW	$\frac{\rho}{B_{REF}}$	$\frac{\rho}{B_{REF}} \frac{C_{REF}}{MW}$	$\frac{\rho}{B_{REF}} \frac{A_{REF}}{\rho_{15}}$

TABLE 2 - COMPENSATION MODES

8.4 COPYING TAGS FROM EXISTING OBJECTS

The **Copy Tags** page presents in the left table (Defined in...) the existing tags and in the right table the tags defined in the object specified as "Origin".

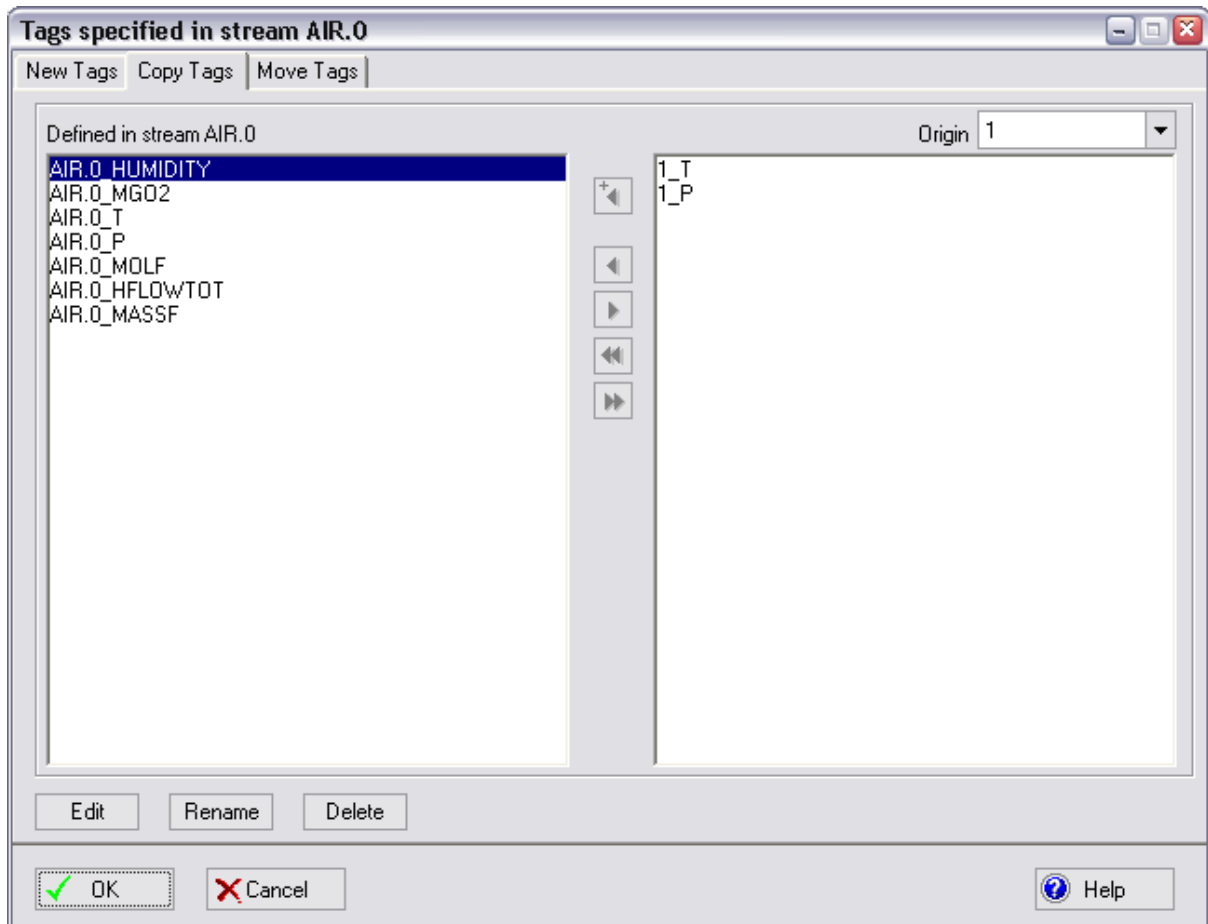


FIGURE 32 - THE COPY TAG DEFINITIONS DIALOG BOX

- select the object from which you want to copy the tags
- select the tags to be copied (multiple select is supported)
- press the  button or press the  button to edit the copied tag

Default names will automatically be given to the copied tags. If needed, you can then rename them with the **Rename** button.

You can use the other buttons, to adjust the selection of tags to be copied:

-  to copy all tags
-  to remove a tag that has been copied
-  to remove all tags that have been copied

8.5 MOVING TAGS FROM EXISTING OBJECTS

The **Move Tags** page presents in the upper left table (Defined in...) the existing tags and in the upper right table the tags defined in the object specified as "Origin".

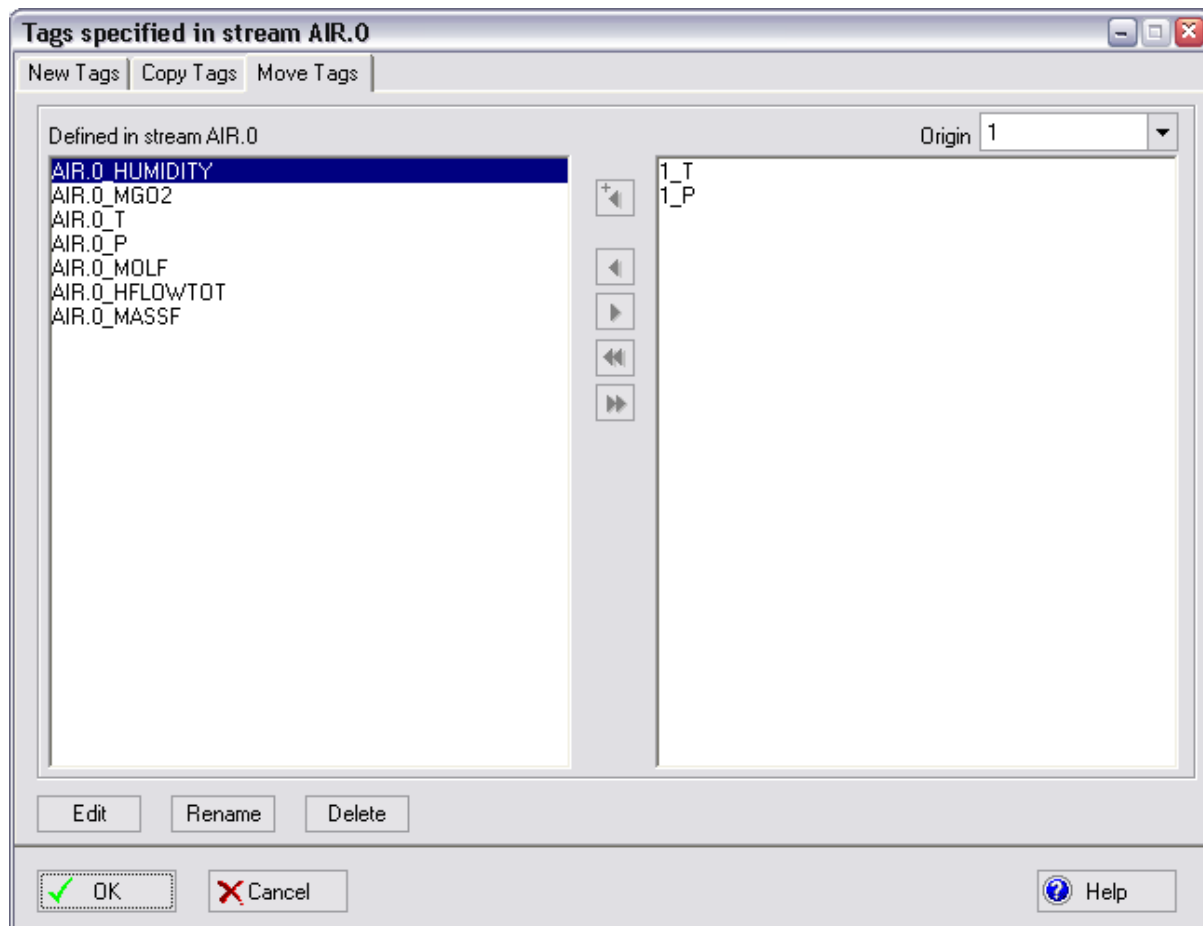


FIGURE 33 - THE MOVE TAG DEFINITIONS DIALOG BOX

- select the object from which you want to move the tags
- select the tags to be moved (multiple select is supported)
- press the  button OR press the  button to edit the moved tag

You can use the other buttons, to adjust the selection of tags to be moved:

-  to move all tags
-  to remove a tag that has been moved
-  to remove all tags that have been moved

8.6 VIEWING AND EDITING MEASUREMENTS VALUES

The **Tools|Measurements** command displays the list of tags that are currently defined in the model.

When in **Edit mode**, the **New**, **Copy**, **Move**, **Edit**, **Rename** and **Delete** buttons allow to modify the definitions of the tags as described here above. You can also **Edit** a tag by double-clicking its name.

The **Details** button toggles the display of the tags details in the bottom part of the screen freeing more room to display additional tags in the main table.

The **Mask** combo box allows you to display only tags where filters, compensations or bounds are defined.

The fields on the second line of the window are masks that you can use to reduce the number of tags that are presented in the table (in the example below only the temperature measurements are shown):

- Name: this mask applies on the tags name
- Type: the type of the tags (pressure, temperature, etc.)
- PFD: the name of the PFD containing the object
- Strem/Unit: the name of the object where tags are specified

You can also click on the column headers to change the order in which the tags are listed

THE MEASUREMENTS PAGE

This is the main page where you can define and display the values associated to a tag. The "Data" combo box is used to switch between Default, Measured and Used values.

Tags specified in project MAIN

Measurements | Definition | Bounds | Filters | Compens | KPI | Historian

Mask: None | Data: Default | Number of tags displayed: 39/231

Name: | Type: T | PFD: | Stream/Unit:

Tagname	Phys. Unit	Default	Accuracy	Type	Reconciled	Accuracy	Penalty	Gain (%)	B	F	C
FW.10_T	C	349.2	1	A	348.038	0.712941	1.34922		☒	☐	☐
VAP.1_T	C	357	1	A	358.392	0.492906	1.93846		☒	☐	☐
VAP.3_T	C	360	OFF	A	358.392	0.492906	0		☐	☐	☐
COAL_T	C	20	2	A	19.9922	1.99999	1.5235...		☒	☐	☐
AIR.2_T	C	339.642	2	A	338.315	1.79377	0.440385		☒	☐	☐
FUMES.1_T	C	2281	OFF	A	2220.79	7.80431	0		☒	☐	☐
FUMES.4_T	C	862.1	5	A	868.181	3.74636	1.47932		☒	☐	☐
FW.9_T	C	212	OFF	A	211.761	0.866321	0		☒	☐	☐
FUMES.5_T	C	359	5	A	361.99	2.24245	0.357519		☒	☐	☐

Stream: FW.10 | T | Plant: VALI_PLANT

Physical Units: C | Measured: 349.2 | Accuracy: 1 | Type: A | Failure Probability: 5 %

Defaults: C | Measured: 349.2 | Accuracy: 1 | Type: A | Failure Probability: 5 %

Bounds: SL -273.2 | LL -273.2 | L -273.2 | H 5000 | HH 5000 | SH 5000

Comment:

New Copy Move Edit Rename Delete Details <<

OK Cancel Help

FIGURE 34 - THE MEASUREMENTS PAGE OF TAGS WINDOW

THE DEFINITION PAGE

You can update here the failure probability and the bounds control settings.

Bound Control is used to activate bound control mechanism on one or both bounds:

- NONE: no action is taken
- SL: bound control on the lower solver bound
- SH: bound control on the upper solver bound
- BOTH: bound control on the both solver bounds

Criteria is the minimum distance at which the variable must be kept with respect to the bound. It can be specified as a **value** or as **AUTO** in which case, the minimum distance is automatically selected according to the type of variable.

Material Stream: 0931P-12 From: 931P-12 / Z93-SRU To: 931P-13 / Z93-SRU

General | Composition | Tags... | Measurements | Definition | Bounds | Filters | Compens | KPI | Historian

Mask: None Number of tags displayed: 1/1

Tagname	Object	Type	Parameter	Fail. Prob.	Bound Control	Criteria
▶ 0931P-12_MASSF	0931P-12	MASSF		5	SL	AUTO

Stream: 0931P-12 MASSF Plant: CDU-PI

Physical Units: Measured Accuracy Type Failure Probability

Defaults: t/h 0 OFF % 5 %

Bounds SL 0 LL 0 L 0 H 3.6e+010 HH 3.6e+010 SH 3.6e+010

Comment:

New Copy Move Edit Rename Delete Details <<

OK Cancel Help

FIGURE 35 - THE DEFINITION PAGE OF TAGS WINDOW

THE BOUNDS PAGE

This page enables to modify the bounds. 3 levels are considered for both high and low bounds:

- warning (L -> H): a warning message will be issued if the measured or the validated values are outside this range
- error (LL -> HH): an error message will be issued if the measured or the validated values are outside this range. (ValiFilter uses these bounds)
- solver (SL -> SH): a fatal error occurs if the measured value is outside this range. The validated value will be kept between these bounds

Tags specified in project MAIN

Measurements | Definition | Bounds | Filters | Compens | KPI | Historian

Mask: None Number of tags displayed: 39/231

Name: Type: T PFD: Stream/Unit:

Tagname	Phys. Unit	SL	LL	L	H	HH	SH	B
FW.10_T	C	-273.2	-273.2	-273.2	5000	5000	5000	☒
VAP.1_T	C	-273.2	-273.2	-273.2	5000	5000	5000	☒
VAP.3_T	C	-273.15	-273.15	-273.15	5000.1	5000.1	5000.1	☐
COAL_T	C	-273.2	-273.2	-273.2	5000	5000	5000	☒
AIR.2_T	C	-273.2	-273.2	-273.2	5000	5000	5000	☒
FUMES.1_T	C	-273.2	-273.2	-273.2	5000	5000	5000	☒
FUMES.4_T	C	-273.2	-273.2	-273.2	5000	5000	5000	☒
FW.9_T	C	-273.2	-273.2	-273.2	5000	5000	5000	☒
FUMES.5_T	C	-273.2	-273.2	-273.2	5000	5000	5000	☒

Stream: FW.10 Type: T Plant: VALI_PLANT

Physical Units: C Measured: 349.2 Accuracy: 1 Type: A Failure Probability: 5 %

Defaults: C Measured: 349.2 Accuracy: 1 Type: A Failure Probability: 5 %

Bounds SL: -273.2 LL: -273.2 L: -273.2 H: 5000 HH: 5000 SH: 5000

Comment:

New Copy Move Edit Rename Delete Details <<

OK Cancel Help

FIGURE 36 - THE BOUNDS PAGE OF TAGS WINDOW

THE FILTERS PAGE

This page defines the Filter settings (a ValiFilter license required) as described in paragraph Filters Settings above.

Tags specified in project MAIN

Measurements | Definition | Bounds | **Filters** | Compens | KPI | Historian

Mask: Number of tags displayed:

Name: Type: PFD: Stream/Unit:

Tagname	Phys. Unit	LL	HH	Value	Accuracy	Type	Fail. Prob.	Cut Off	F
FW.10_T	C	-273.2	5000	NONE	NONE	%	NONE	NONE	☐
VAP.1_T	C	-273.2	5000	NONE	NONE	%	NONE	NONE	☐
VAP.3_T	C	-273.15	5000.1	NONE	NONE	%	NONE	NONE	☐
COAL_T	C	-273.2	5000	NONE	NONE	%	NONE	NONE	☐
AIR.2_T	C	-273.2	5000	NONE	NONE	%	NONE	NONE	☐
FUMES.1_T	C	-273.2	5000	NONE	NONE	%	NONE	NONE	☐
FUMES.4_T	C	-273.2	5000	NONE	NONE	%	NONE	NONE	☐
FW.9_T	C	-273.2	5000	NONE	NONE	%	NONE	NONE	☐
FUMES.5_T	C	-273.2	5000	NONE	NONE	%	NONE	NONE	☐

Stream: Plant:

Physical Units: Measured: Accuracy: Type: Failure Probability: %

Defaults: %

Bounds SL: LL: L: H: HH: SH:

Comment:

New Copy Move Edit Rename Delete Details <<

OK Cancel Help

FIGURE 37 - THE FILTERS PAGE OF TAGS WINDOW

THE COMPENSATIONS PAGE

This page defines the Compensation settings as described in paragraph Compensation Settings above.

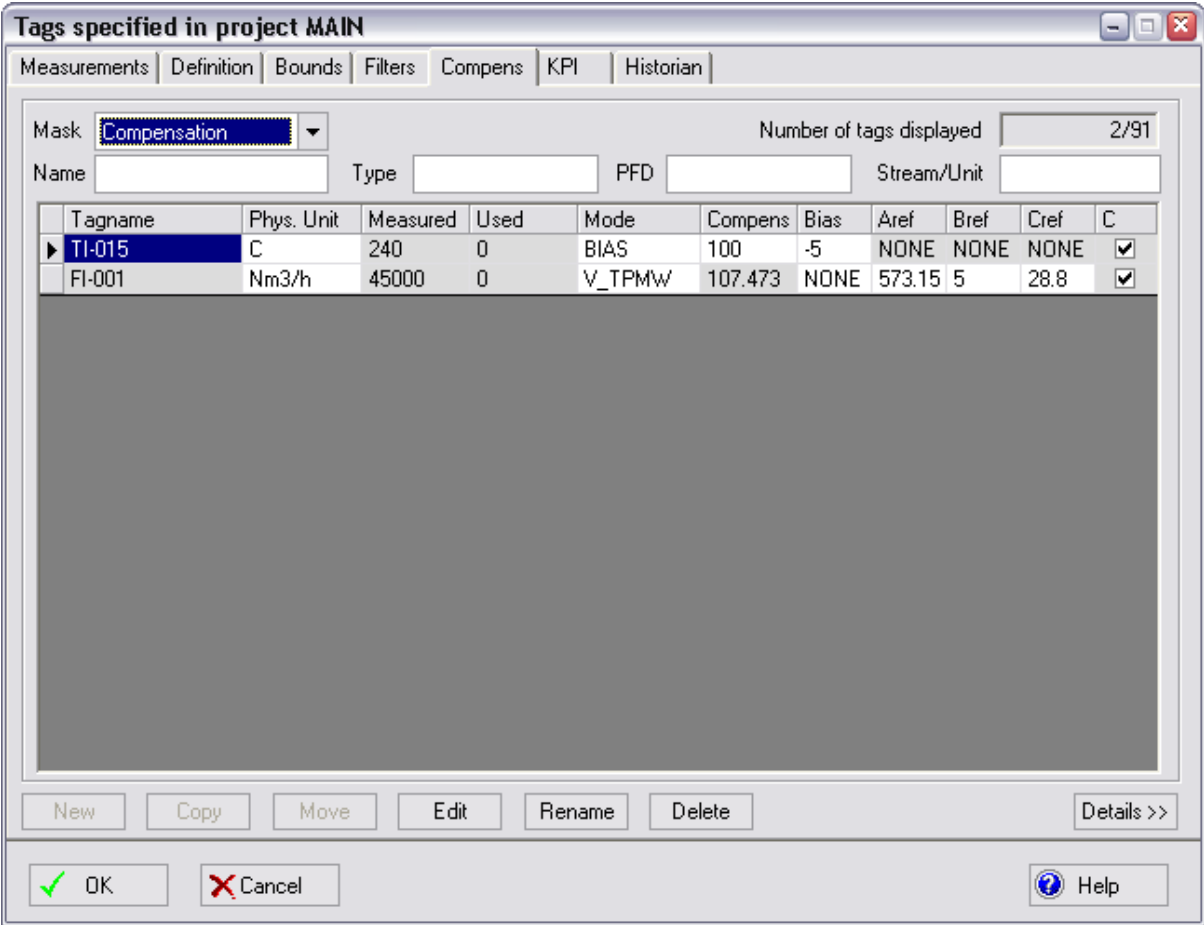


FIGURE 38 - THE COMPENS PAGE OF TAGS WINDOW

THE KPI PAGE

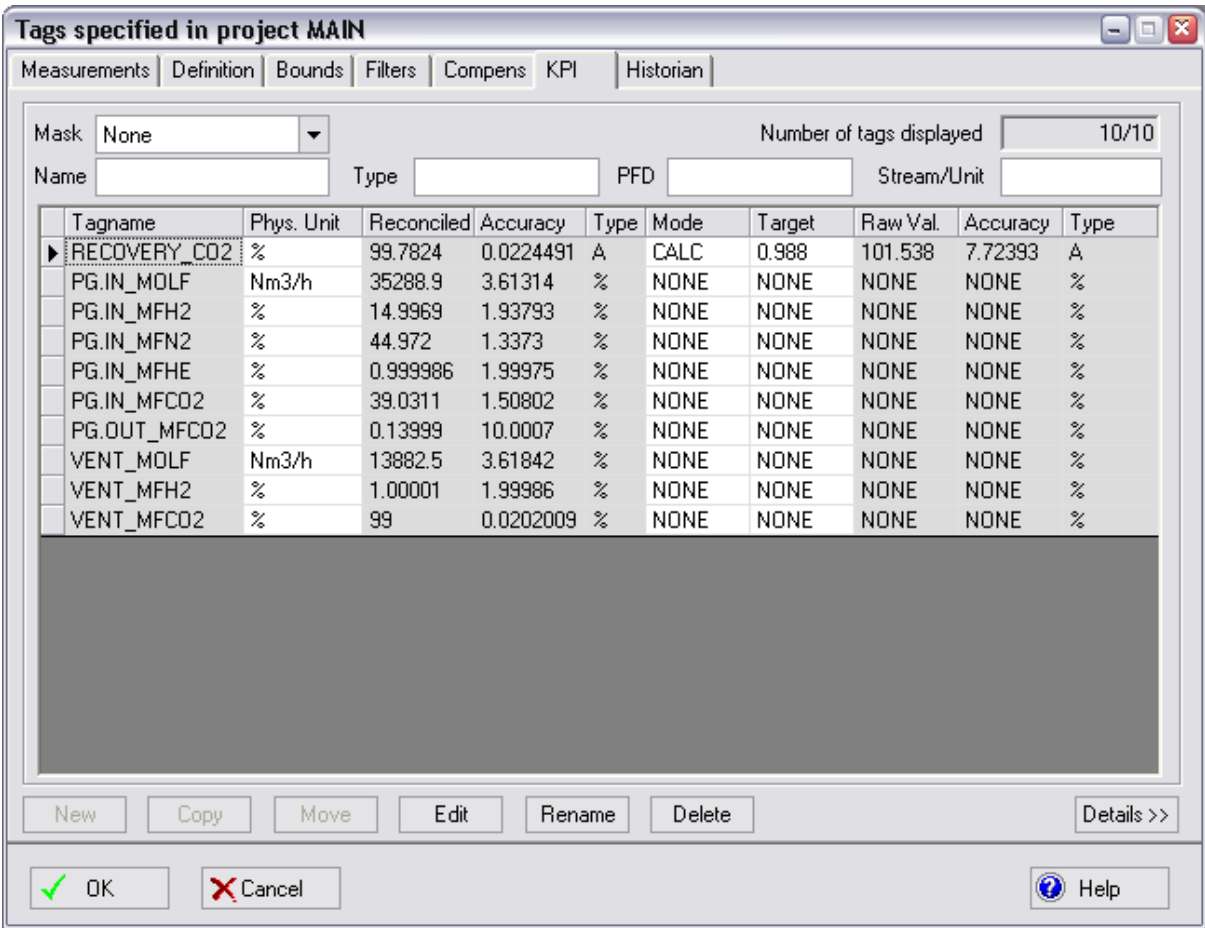


FIGURE 39 - THE KPI PAGE OF TAGS WINDOW

Tags defined as KPI (Mode= "ON" or "CALC") are gathered in a specific table in the html report. The target value is only used as an indication of the KPI expected value.

Setting the mode as "CALC" indicates you want ValiEngine to calculate what would be the value and accuracy of the KPI when calculated on basis of the measured values. This mode can only be used for tags calculated through a Flex code involving only measured values (using the function CALL GTVARTAG (...)).

This enables to quantify the benefit of the reconciliation on the correctness and uncertainty of the KPI.

KPI						
Name	Rec. val.	Rec. acc.	Raw val.	Raw acc.	Target	Unit
RECOVERY_CO2	99.782	0.224E-01	101.54	7.72	0.98800E-08	%

FIGURE 40 - THE KPI TABLE IN THE HTML REPORT

THE HISTORIAN PAGE

The Historian page defines tags settings related to ValiLink and ValiDatabase:

- Plant: Plant defining the direct access historian server
- DB name: Name of the tag in the historian server
- ReadDB: Aggregation for the tag value to be acquired from the historian:
 - None: no acquisition
 - Average: average over the validation time window
 - BeginDate: value at the beginning of the validation time window
 - EndDate: value at the end of the validation time window
 - Min: minimum value over the time window
 - Max: maximum value over the time window
 - Mean: arithmetic average of all the stored values over the time window
- WriteDB: Tags to be written back to the historian:
 - None: no value returned
 - Suffix_V: the validated value will be returned to the historian, adding the suffix _V to the tag name
- ValiDB: Check-box to select the tags to be exported to ValiDatabase through the arc file
- Aggregation: consolidation mode for ValiReport:
 - AVG: average over the validation time window
 - Sum: sum over the validation time window
 - Begin: value at the beginning of the validation time window
 - End: value at the end of the validation time window

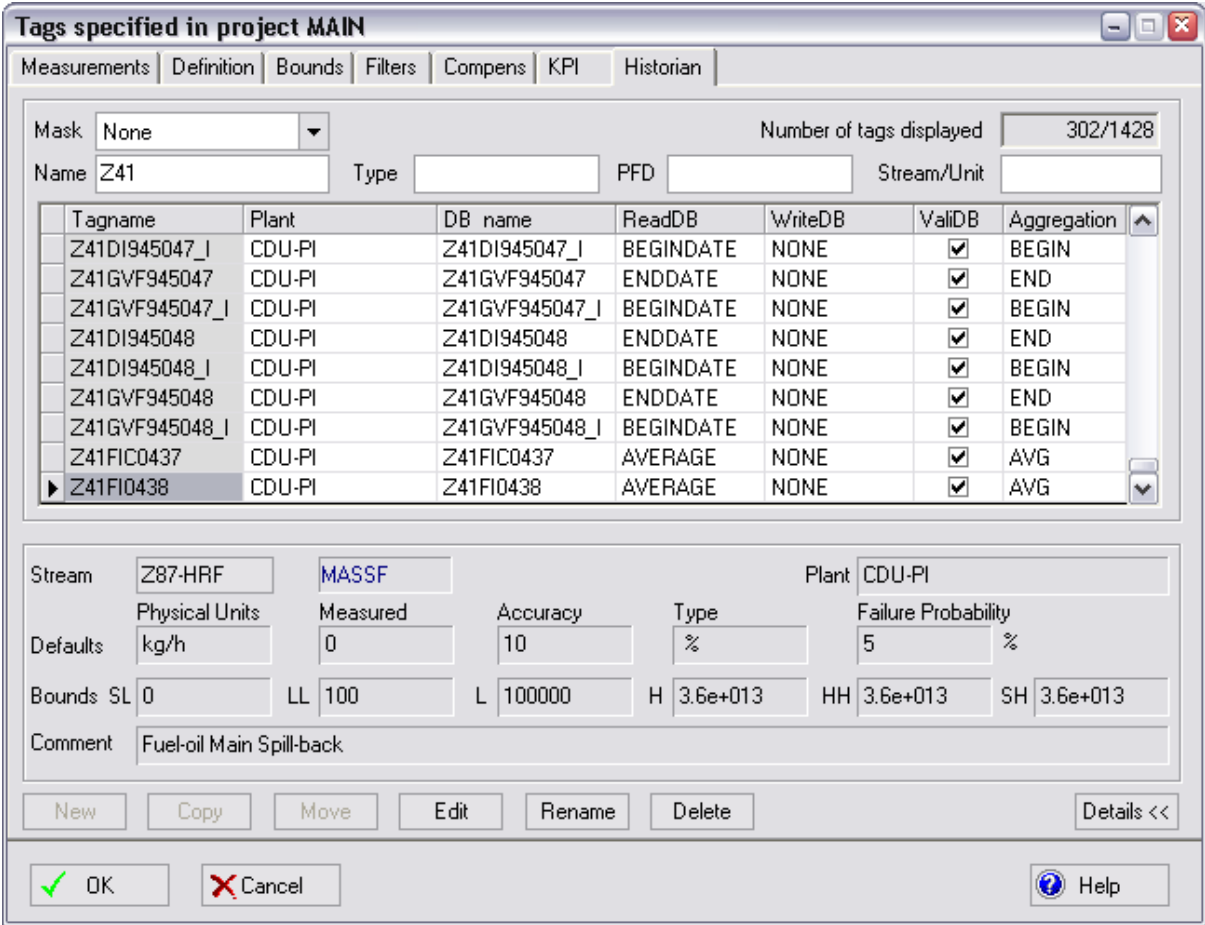


FIGURE 41 - THE HISTORIAN PAGE OF TAGS WINDOW

8.7 USING CONTROL STRINGS

INTRODUCTION TO CONTROL STRINGS

The Control Strings are used to carry out various actions depending on the measured, and sometimes validated value of a given Tag. The syntax is:

&1 "STRING1" to define a control string for messages

&2 "STRING2" to define a control string for the measured value

&3 "STRING3" to define a control string for the accuracy

The three control strings have the same syntax:

ACTION1 (CONDITION1), ACTION2 (CONDITION2), ETC.

The string is processed sequentially. If Condition1 is satisfied then Action1 is taken. The second pair Action2(Condition2) is then normally processed. If Condition2 is met then Action2 is taken, etc. Actions defined without Condition are always performed.

The CONDITION can be:

- (<number): satisfied if measured value < number (Number can also be SL,LL, or L)
- (>number): satisfied if measured value > number (Number can also be SH, HH or H)
- (number): satisfied if measured value = number $\pm e$ (e is taken as the maximum of 1.10^{-6} and $|measurement|.10^{-6}$)
- (num1,num2): satisfied if num1 < measured value < num2
- (OFF): satisfied if measurement is off
- (CST): satisfied if measurement is considered constant

The ACTION commands depend on the type of control string.

CONTROL STRINGS FOR MESSAGES

Message control strings are processed twice:

- before the resolution, using measured values to evaluate the Condition
- after the resolution, using validated values to evaluate the Conditions

The ACTION can be:

- 'message': print message in <model>.r1v and <model>.r4v files
- OK: stop processing string. This interrupts the normal sequential processing of the string. Further conditions are no more evaluated and their actions are ignored.

Example:

```
&1 "'VERY LOW' (<380) , 'LOW' (380,400) , 'HIGH' (460,480) , 'VERY HIGH' (>480) "
```

Assuming the measured value of this Tag (named TI-001) is 378.4 °C and the validated value is 383.2 °C.

- The message: "TI-001 (378.4 C) : VERY LOW" is printed at the beginning of the files
- The message: "TI-001 (383.2 C) : LOW" is printed at the end of the files

CONTROL STRINGS FOR MEASURED VALUE

Control strings for measured value are processed before the resolution. The conditions are processed by using the measured value, and when a condition is satisfied, its associated action is activated and modifies the used value of the Tag.

The ACTION can be:

- number: set measurement value to number
- DEF: set measurement value to the default value (as defined in the model)
- VAL: restore original measured value
- CUTOFF: the stream or unit is set OFF
- OK: stop processing the string. This interrupts the normal processing of the string. Further conditions are no more evaluated and their actions are ignored

Example:

```
&2"DEF(<380),400(380,400),460(460,480),DEF(>480)"
```

This would replace the measured value by the default value whenever the measurement value is below 380 or above 480. If the measured value lies between 380 and 400, it will be replaced by 400. If it lies between 460 and 480, it will be replaced by 460.

If the measured value of this Tag (named TI-001) is 378.4 °C, the default value of the Tag TI-001 will be used in place of the actual value.

CONTROL STRINGS FOR ACCURACY

Control strings for accuracy are processed before the resolution. The conditions are processed using the measured value of the Tag, and when a condition is satisfied, its associated action is activated and modifies the used accuracy.

The ACTION can be:

- number: set the accuracy to number (a negative number corresponds to a relative precision expressed in %)
- DEF: set accuracy to the default value
- VAL: restore original accuracy
- OFF: turn measurement off
- CST: consider measurement as a constant
- OK: stop processing string. This interrupts the normal sequential processing of the string. Further conditions are no more evaluated and their actions are ignored

Example:

```
&3"OFF(<380),50(380,400),50(460,480),OFF(>480)"
```

This will turn off the measurement if its measured value lies outside the range 380 to 480. It also puts an accuracy of 50 if the measured value lies between 380 and 400 or 460 and 480.

An equivalent syntax would be: `&3"OFF,50(380,480),VAL(400,460)"`

The measurement is first turned off. Then, if the measured lies between 380 and 480, it is turned on

again with accuracy of 50. Finally, if the measured value lies between 400 and 460, the measured accuracy is restored.

A third equivalent syntax would be: `&3"OK(400,460),50,OFF(<380),OFF(>480)"`

The OK action stops the processing of the string if the measured value lies between 400 and 460. Otherwise, it applies a value of 50 to the accuracy and turns the measurement off if it is below 380 or above 480.

Attention: the action OK with no (condition) allows to temporarily de-activate a string without having to retype it when you want to reactivate it: OK,OFF(<380),50(380,400),50(460,480),OFF(>480)

In this case, the action OK is always processed which stops processing the string that is thus de-activated. To reactivate the string, simply remove the OK action.

COLOR HIGHLIGHTING AND TAGS

The Command **Tools | Highlight Measurements** enable to flag the PFDs, UNITS and STREAMS with different colors:

- None/All/Active ones: objects for which tags are defined (and active or not)
- Penalty: objects for which some tags are either significantly corrected, suspicious, eliminated or filtered

The coloring follows the following order: eliminated, filtered, suspicious, significant, measurement active or existing. For example, a tag that has been filtered and eliminated will be colored as eliminated.

The coloring system for penalty automatically propagates through PFDs containing the corresponding objects.

SECTION 9: THE THERMODS

Each material STREAM refers to a THERMOD, which indicates:

- the list of chemical components that can, but not necessarily do, appear in the STREAM
- the thermodynamic methods that must be used for various calculations
- depending on the thermodynamic methods, binary interaction parameters

9.1 ACCESSING THE THERMODS NOTEBOOKS

To access the notebook of a THERMOD, double-click it in the Browse Tree or use the Find facility

The THERMOD Notebook has three active pages:

- Compounds: to modify the list of compounds of the THERMOD
- Methods: to modify the methods used for calculating STREAM properties
- Binary Parameters: to modify the binary parameters of the THERMOD.

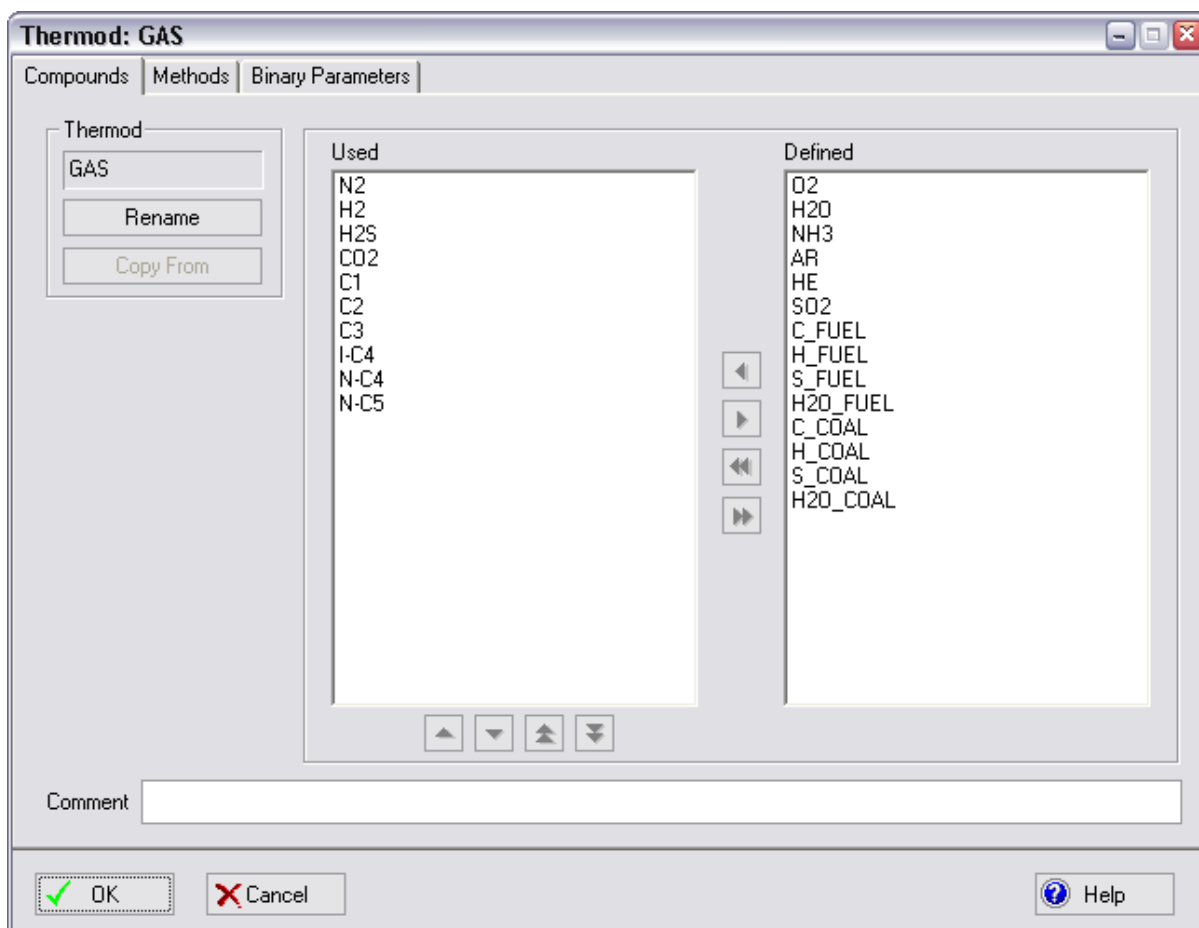


FIGURE 42 - THE COMPOUNDS PAGE OF A THERMOD NOTEBOOK

9.2 MODIFYING THERMODS

You modify the list of **COMPOUNDS** by transferring compounds between the "Used" list and the "Defined" list through the use of the     buttons. Multiple select is supported.

Use the     to reorder the compounds.

You modify the **Methods** by using the drop-down list boxes for each phase property. The default method button allows to reset the default methods.

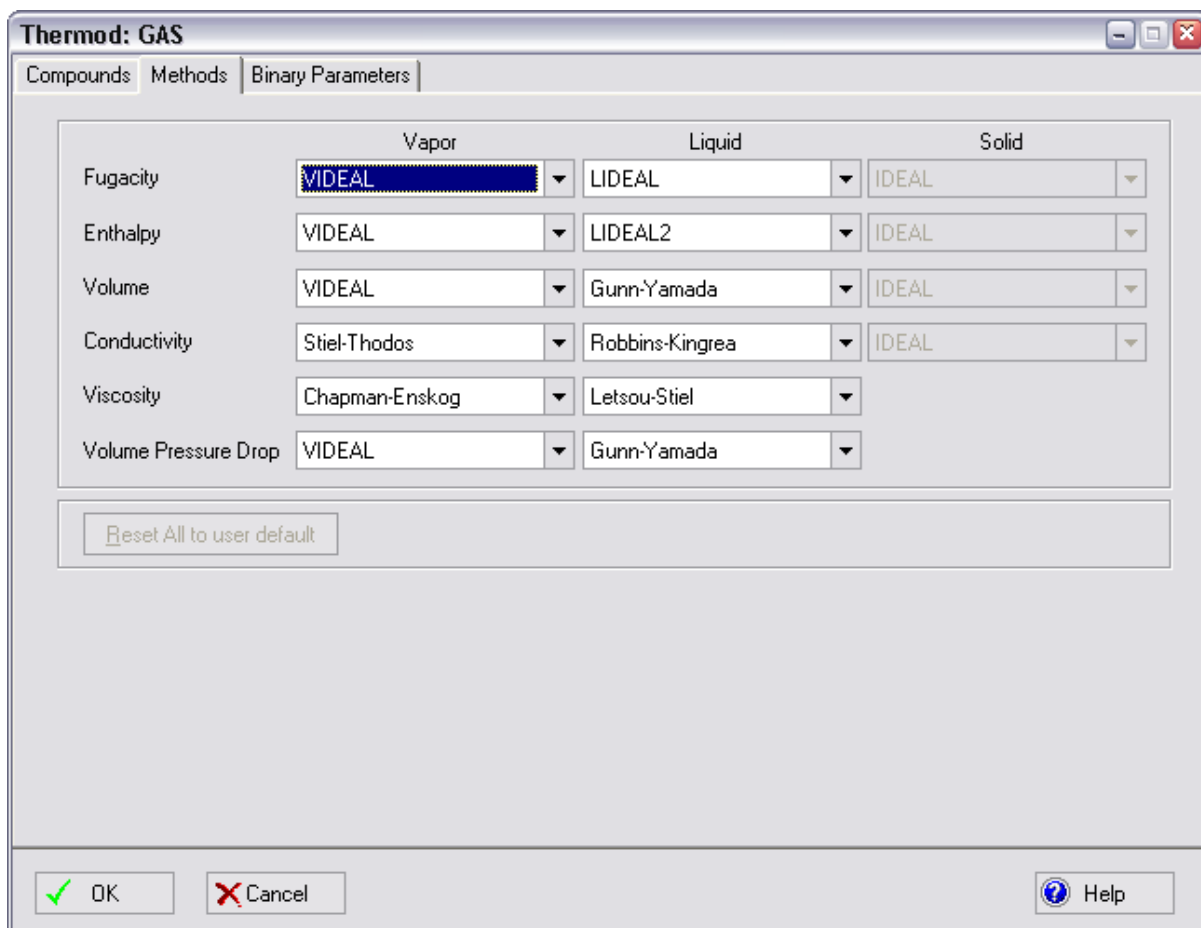


FIGURE 43 - THE METHOD PAGE OF A THERMOD NOTEBOOK

To add new binary parameters, select the compounds, the method and the parameter, specify a value and press the Add button.

The values of existing binary parameters can be changed directly in the table.

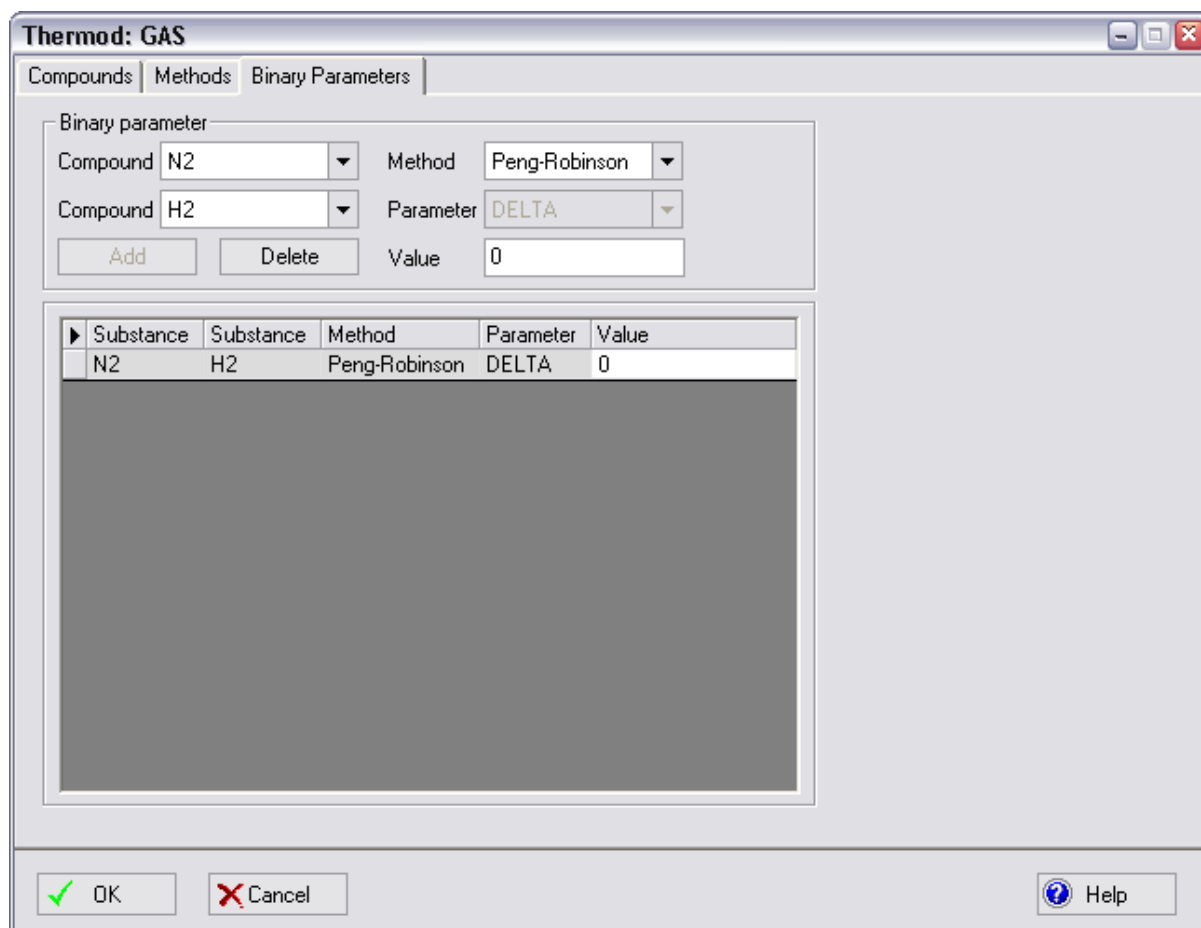


FIGURE 44 - THE BINARY PARAMETERS PAGE OF A THERMOD NOTEBOOK

9.3 CREATING THERMODS

To create a new THERMOD, select the Edit | Add THERMOD menu command.

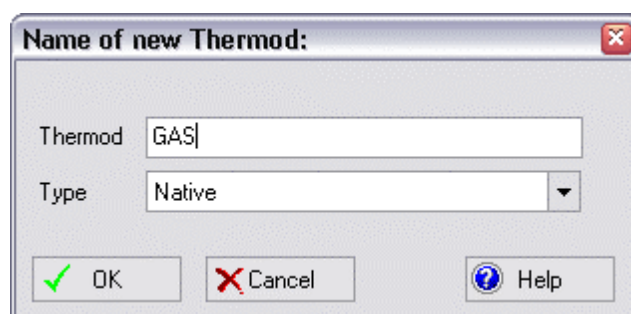


FIGURE 45 - CREATING A THERMOD

The compounds page of the THERMOD notebook is then displayed. You can use the Copy From button to copy the list of compounds, methods and binary parameters from another THERMOD.

9.4 RENAMING THERMODS

You can rename a THERMOD via the **Rename** button of its notebook or by right-clicking in the Browse Tree and selecting the **Rename** command.

9.5 DELETING THERMODS

To delete a THERMOD, right-click it in the Browse Tree and select the **Delete** command.

Attention: you can only delete unused THERMODS. If some STREAMS are referring to this THERMOD, then it cannot be deleted.

SECTION 10: THE COMPOUNDS

COMPOUNDS data can be accessed via the COMPOUNDS section of the Browse Tree.

The **General** page of the COMPOUNDS notebook lets you review the data associated with the compounds. Changing the values is however prohibited.

Compound: AR

General

Compound: AR

Rename

Copy From

Longname: ARGON

Epic: 1 Abstract: 0 Origin: System

Formula: AR1 Family:

Name	Value	Physical Unit	Description	Origin
HOF	0	kJ/kmol	Standard enthalpy of formation	DATA
GOF	0	kJ/kmol	Gibbs s std. energy of formation	DATA
MOLW	39.948	kg/kmol	Molecular weight	DATA
TB	87.293	K	Boiling temperature	DATA
TC	150.75	K	Critical temperature	DATA
PC	48.7373	bar	Critical pressure	DATA
VC	0.07495	m3/kmol	Critical volume	DATA
OMEGA	-0.002		Pitzer's acentric factor	DATA
CPLA	-79.8392		Liquid Heat Capacity (kJ/kmol/K)	DATA
CPLB	3.87165		CPL = A + B*T + C*T² + D*T³	DATA
CPLC	-0.0425263		.	DATA
CPLD	0.000162536		.	DATA
TMIN(PV)	61.11	K	Temperature range for vapor pressure	DATA
TMAX(PV)	147.784	K	.	DATA

Comment:

OK Cancel Help

FIGURE 46 - THE GENERAL PAGE OF THE COMPOUND NOTEBOOK

10.1 ADDING COMPOUNDS TO A MODEL FILE

To add new compounds to the model, select the **Edit|Add COMPOUND** command.

Before adding a compound, you can first modify the default list of Available COMPOUNDS:

- select another library of compounds. Five standard libraries are available:
 - Fluids.som: most common fluid compounds
 - Fluids.all: fluid compounds (complete Belsim data base)
 - Solids.all: solid compounds
 - Fuels.all: pseudo-compounds for fuel-oil and coal
 - Petrol.all: pseudo-compounds for petroleum fractions

Depending on your installation, one or several additional data bases might be available.

- limit the number of available compounds by selecting a given Family (e.g. Paraffins or Sulphur derivatives) in the Family field.
- limit the number of available compounds by entering a given text in the Mask field. Only the compounds which include that string will be shown.
- modify the order in which the compounds are shown, by checking one of the check boxes above the list. By default, compounds are sorted by their chemical formula.

Add Compound

Compound library: fluids.som
 Family: <All>
 Mask:
☐ Name ☒ Formula ☐ LongName

Compound	Formula	LongName
AR	AR1	Argon
12DEB	C10H14	12-Diethylbenzene
13DEB	C10H14	13-Diethylbenzene
N-C10	C10H22	N-Decane
NAPHT	C10H8	Naphthalene
DPHNYL	C12H10	Diphenyl
DPHE	C12H10O1	Diphenylether
ANTHRA	C14H10	Anthracene
C1CL3	C1H1CL3	Trichloromethane
HCN	C1H1N1	Hydrogen Cyanide
C1CL2	C1H2CL2	Dichloromethane
C1AL	C1H2O1	Formaldehyde
HFORMIC	C1H2O2	Formic Acid
C1CL	C1H3CL1	Chloromethane

Compound	Epic	Formula	Longname	Abstract	Family
AR	1	AR1	ARGON	7440371	Miscellaneous

Comment:

OK Cancel Help

FIGURE 47 - THE ADD COMPOUND NOTEBOOK

To add a compound:

- click on one of the compounds shown in the list of Available compounds
- press the **Load** key

Attention: if you want to compound with the same name as an existing one, you need first to rename that compound.

10.2 REMOVING COMPOUNDS FROM A MODEL FILE

Only compounds that are not referenced in any REACTION or THERMOD can be removed. To remove a compound:

- right click on one of the compounds shown in the Browse Tree
- select the **Delete** command

10.3 RENAMING COMPOUNDS IN A MODEL FILE

To rename a compound:

- right click on one of the compounds shown in the Browse Tree
- select the **Rename** command
- check the **auto-rename tag definitions** box to automatically rename the tags associated with the compound
- enter its new name
- press the **OK** button

SECTION 11: THE REACTIONS

Reactors of type RCTVAL, RLVVAL, etc. refer to REACTIONS. Each REACTION is a separate object and has:

- a name
- a comment
- a stoichiometry: list of reactants and products with stoichiometric coefficients
- further parameters describing the kinetics of the reaction (only used by RKIVAL unit)

Attention: REACTIONS are by default considered as reversible by VALI (when no kinetics is involved). REACTIONS can be set as irreversible by setting adequate bounds on the corresponding reaction extents variables. In this case you must use the new SQP solver.

11.1 ACCESSING REACTIONS NOTEBOOK

To access the notebook of a REACTION, double-click the reaction in the Browse Tree. Alternatively you can also access the REACTIONS notebooks through the **Find** facility.

The REACTION Notebook has two active pages:

- General: to modify the REACTION name, its comment and the list of reactants and products and their stoichiometric coefficient
- Parameters: to modify the kinetic parameters of the reactions. (These parameters are only used by RKIVAL units)

Reaction: SYNT

General Parameters

Reaction: SYNT

Rename Copy From

Reactants

Name	Coefficient	Order
H2	3	0
N2	1	0

Compounds

AR
CO2
H2O

Products

Name	Coefficient	Order
NH3	2	0

Equation: 3 H2 + 1 N2 <=> 2 NH3

Comment: N2 + 3 H2 <-> 2 NH3

OK Cancel Help

FIGURE 48 - THE GENERAL PAGE OF THE REACTIONS NOTEBOOK

11.2 MODIFYING THE STOICHIOMETRY OF REACTIONS

To change the stoichiometry of a reaction:

- transfer compounds between the list of available compounds, reactants and products with the and buttons
- modify the stoichiometric coefficient by changing the value in the coefficient field

The **equation** with the coefficients and compounds is automatically updated. At your convenience you can copy and paste it in the comment field of the reaction.

Use the , , , buttons to reorder the variables.

11.3 MODIFYING THE KINETICS OF REACTIONS

To modify the kinetics of a reaction:

- activate the **Parameters** page of the REACTION Notebook
- modify the various parameters
- press the **OK** button

Name	Description	Value	Physical Unit	Status
KA	Constant 1 for equilibri...	-1		Input
KB	Constant 2 for equilibri...	-1000	C	Input
ETYPE	Equilibrium Method	1		Input
KD	Kinetic Constant	-1		Input
ACT.EN	Activation Energy	-1	kJ/kmol	Input
VTYPE	Velocity Method	1		Input
DHR	Heat of Reaction	-1	kJ/kmol	Input
TREF	Reference Temperature	-1000	C	Input

Default Values Parameters

OK Cancel Help

FIGURE 49 - THE REACTION PARAMETERS NOTEBOOK

The description of the kinetics of a reaction is only taken into account by the RKIVAL unit. This description can be partially overwritten by a Flex code attached to the unit (see section "[Using Flex for defining kinetics](#)"). The actual reaction velocities are defined as the product of two terms: a velocity term V_i and an equilibrium term E_i :

$$R_i = V_i \cdot E_i$$

The equilibrium term E_i is expressed as:

$$E_i = \left(1 - \frac{1}{K_i} \frac{\prod_{j=1}^n X_j^{b_j}}{\prod_{j=1}^n X_j^{a_j}} \right)$$

Where:

- a_j and b_j are respectively the stoichiometric coefficient of the reactants and of the products
- K_i is the equilibrium constant of the reaction
- X_j signification depends on the value of the parameter ETYPE:
 - ETYPE=1: X_j is the fugacity. The equilibrium constant is calculated from compound properties.
 - ETYPE=2: same as 1, but the fugacity is calculated at the reactor inlet and kept constant afterwards
 - ETYPE=3: X_j is the fugacity. The equilibrium constant is given by the user
 - ETYPE=4: same as 3, but the fugacity is calculated at the reactor inlet and kept constant afterwards
 - ETYPE=5: X_j is the partial pressure. The equilibrium constant is given by the user.
 - ETYPE=6: X_j is the molar fraction. The equilibrium constant is given by the user.
 - ETYPE=7: X_j is the concentration (mole/l). The equilibrium constant is given by the user.
 - ETYPE=8: $E_i = 1$ (No equilibrium term)

When given by the user, the equilibrium constant has the following form:

$$\ln(K_i) = K_{Ai} - K_{Bi} / T$$

Where:

- T is the temperature (expressed in K) and K_{Ai} and K_{Bi} are given by the user

The velocity term V_i is expressed as:

$$V_i = k_i \prod_{j=1}^n X_j^{c_j}$$

Where:

- c_j is the order of the reaction of the different compounds
- k_i is the equilibrium constant of the reaction
- X_j signification depends on the value of the parameter VTYPE:
 - VTYPE=1: X_j is the fugacity
 - VTYPE=2: same as 1, but the fugacity is calculated at the reactor inlet and kept constant afterwards
 - VTYPE=3: X_j is the partial pressure
 - VTYPE=4: X_j is the molar fraction
 - VTYPE=5: X_j is the concentration (mole/l)

The velocity constant has the following form:

$$\ln(k_i) = K_{0i} - \text{ACT.EN}_i / R / T$$

Where:

- T is the temperature (expressed in K), R the ideal gas constant and K_{0i} and ACT.EN_i are given by the user.

The parameters DHR and TREF can be used to specify the heat of reaction in place of calculating it from the heat of formation of the compounds.

Attention: In a given reactor, all reactions must be based on the same type of equilibrium equation and of velocity equation (parameters ETYPE and VTYPE). Some reactions can however be individually specified as type 8 (no equilibrium term).

11.4 CREATING REACTIONS

To create a new REACTION select the **Edit | Add Reaction** Menu command.

You can copy the data from another reaction with the **Copy From** button.

11.5 RENAMING REACTIONS

To rename a REACTION you can either use the **Rename** button in the REACTION notebook or right-click the REACTION in the Browse Tree and select the **Rename** command.

11.6 DELETING REACTIONS

To delete a REACTION, right-click the REACTION in the Browse Tree and select the **Delete** command.

Attention: you can only delete unused REACTIONS. If some UNITS are referring to a given REACTION, then it cannot be deleted.

SECTION 12: HANDLING PHYSICAL UNITS

Each parameter has its own Physical Units, which can be changed individually, via drop down boxes.

12.1 CHANGING PHYSICAL UNITS

Clicking on the **Physical UnitS** section in the Browse Tree or selecting the **Tools|Physical UnitS** menu command opens the Physical UnitS Notebook. In this notebook, you can modify the default physical units for all the UNITS and STREAMS parameters and measurements. You can also modify the number of digits to be shown after the decimal separator when displaying values in the **Result boxes**.

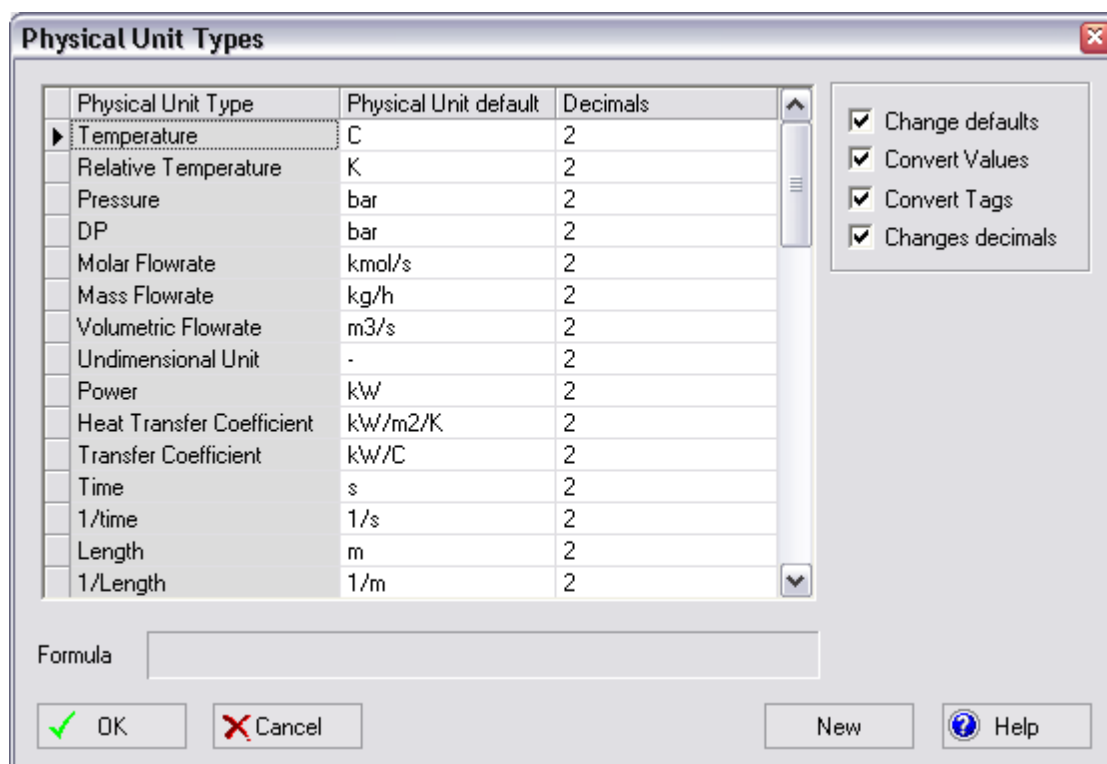


FIGURE 50 - THE PHYSICAL UNIT NOTEBOOK

The table displays the current defaults physical units used for each type of variable, as well as the number of digits for result boxes. Any new object you will create will use the default physical units.

On the right side, a number of check boxes are presented:

- **Change Defaults:** when checked, the new settings will apply to the new objects you will create
- **Convert Values:** when checked, the existing objects (unit and stream parameters) will be converted to the new default physical units, if they are currently expressed in the previous default units
- **Convert Tags:** when checked, the existing tags will be converted to the new default physical units, if they are currently expressed in the previous default units
- **Change Decimals:** when checked, the existing result boxes values will be redisplayed using the new default number of decimals

Attention: When using the **Convert Values**, only the variables that were using the old default physical units will be updated. For example, when changing the default for pressure from bar to atm, the pressures that are defined in bar will be converted in atm, but pressures defined in Pa, mmHg or any other unit will be left unchanged.

12.2 CREATING NEW PHYSICAL UNITS

You can create your own physical units by pressing the New button in the Physical Unit Notebook.

The dialog box titled "Define New Physical Unit" contains the following fields and controls:

- Type:** A dropdown menu currently showing "Mass Flowrate".
- Phys. Unit Name:** A text field containing "[T/sec]".
- Conversion Factor 1:** A text field containing "1000".
- Conversion Factor 2:** A text field containing "0".
- Formula:** The fields are combined into the formula: $[\text{T/sec}] = 1000 * [\text{kg/s}] + 0$.
- Save as User Physical Unit:** An unchecked checkbox.
- Buttons:** "OK" (with a green checkmark icon), "Cancel" (with a red X icon), and "Help" (with a question mark icon).

FIGURE 51 - CREATING NEW PHYSICAL UNIT

Enter the Physical Unit Name and the Conversion Factors to be used to convert the value from the new physical unit to the default physical unit.

Check the **Save as User Physical Unit** check-box to save the settings in the user defaults file (usr/pud.cfg file in VALI installation folder) and share it through different models and users.

User physical units are always saved in the model file as well, to ensure portability through different installations.

SECTION 13: RESULT BOXES

Result boxes can be associated to PFDs units and streams to display their values directly in the flow-sheet.

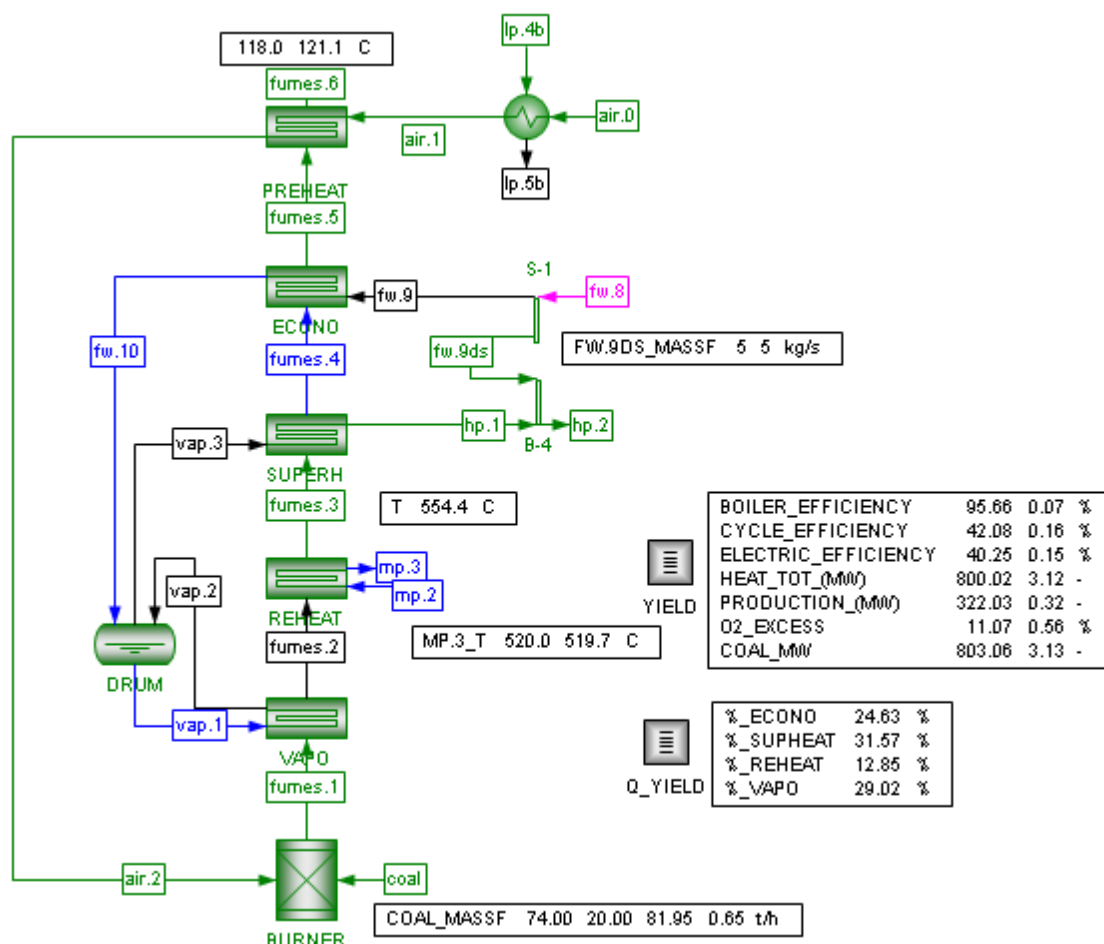


FIGURE 52 - THE RESULT BOXES

13.1 RESULT BOXES NOTEBOOK

The Notebook is similar for PFDs, UNITS and STREAMS. In the example below a Result box linked to a stream is shown.

In any Result box, you can display values related to the object to which the Result box is linked but also values related to any other UNIT or STREAM.

For streams, you can display values related to parameters, composition (partial molar flow rates) and measurements. For units, you can display values related to parameters and measurements.

The left side shows the Result box name, the PFD where it is located as well as the name of the object to which it is linked. When the **link** check box is checked, the result box will be moved together with the object. Otherwise its location is static.

The middle of the notebook ("What to show") indicates what information will be shown for each of the selected variables listed below.

The right side of the notebook ("What to select") enables to add more variables to the result box.

Use the     buttons to add or remove variables to the list of selected variables. Use the     buttons to reorder the variables.

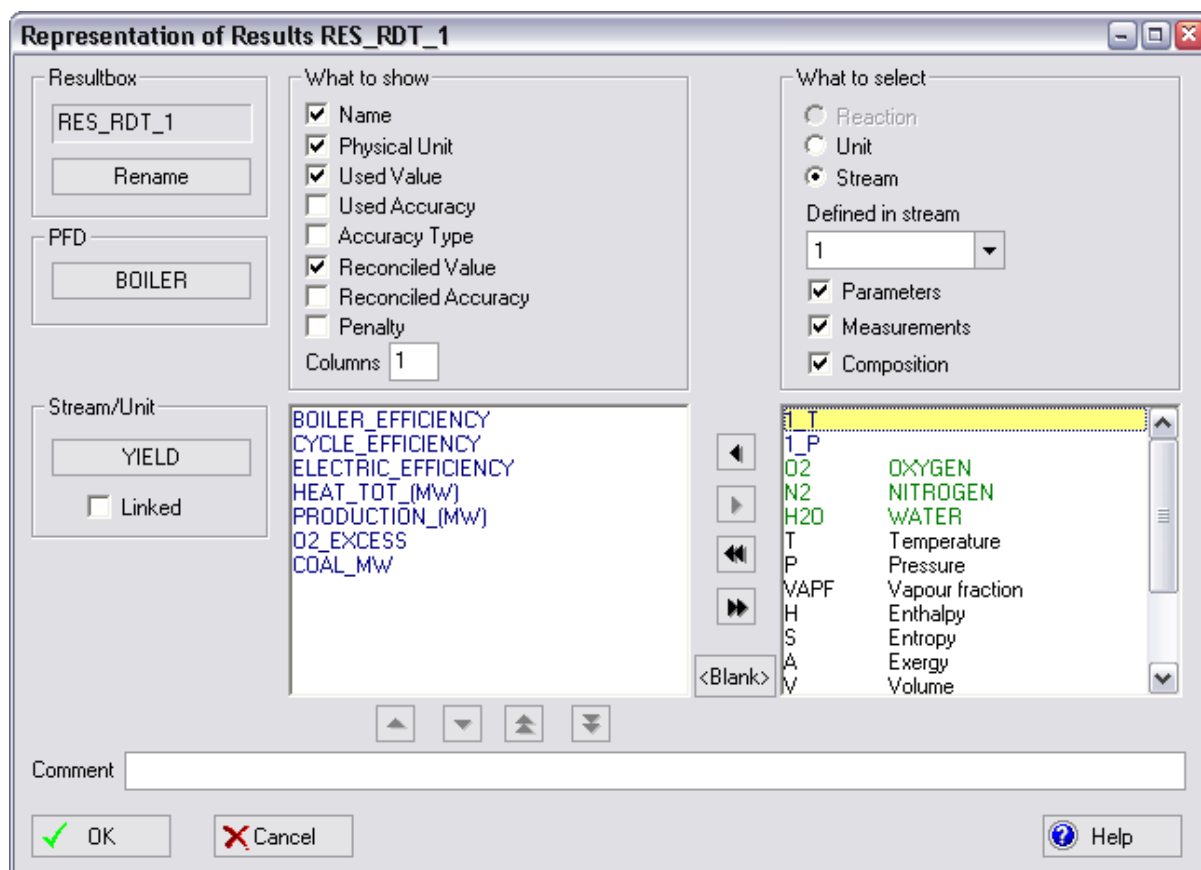


FIGURE 53 - THE RESULT BOX NOTEBOOK

13.2 CREATING RESULT BOXES

To create a new Result box:

- open the PFD where the result box must be created
- right click on a stream, unit or PFD
- select the **Create Results** command
- click at the place where you want to create the Result box
- the Result notebook is then activated
- select the values to be shown
- press the **OK** button

13.3 DELETING RESULT BOXES

To delete a Result box:

- right click the Result box
- select the **Delete** command

or:

- click the Result box
- press the **Del** key

SECTION 14: A PRACTICAL EXAMPLE

Before starting this example, you should be familiar with the basic functioning of ValiModeller, as explained in the preceding sections.

14.1 THE EXAMPLE : A GAS TURBINE

The gas turbine is shown below.

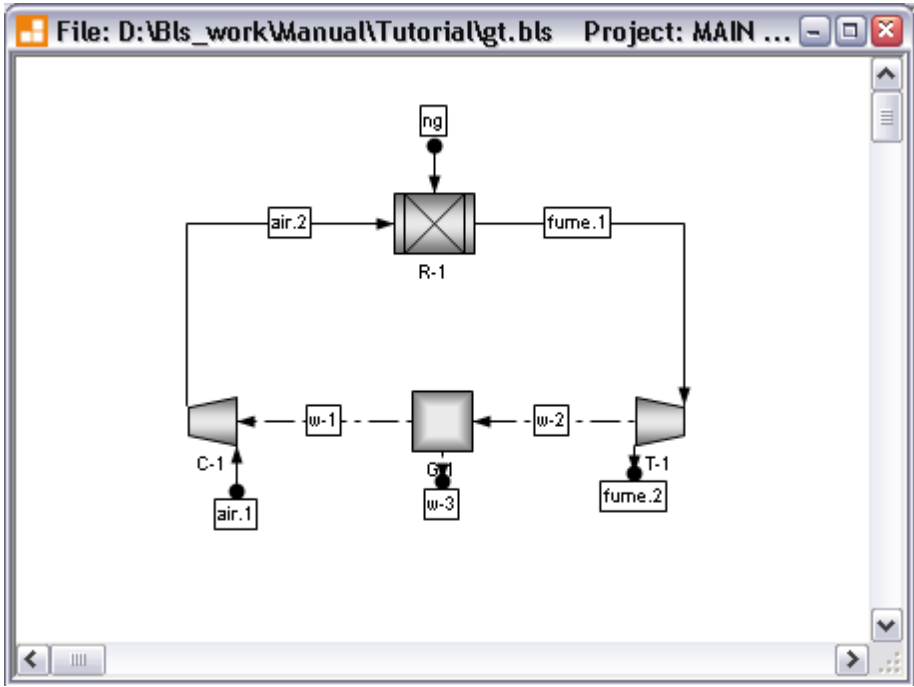


FIGURE 54 - FLOW-SHEET OF THE EXAMPLE

It consists of the following devices :

- a compressor: C-1
- a burner: R-1
- a turbine: T-1
- a generator: G-1

The gas burned will be assumed to be pure methane. The other assumptions are summarized in following table.

UNIT	DATA	VALUE	ACCURACY	PHYSICAL UNITS
R-1	DP	0.1	0.01	bar

C-1	EFFIC	77	5%	%
T-1	EFFIC	80	5%	%

TABLE 3 - ASSUMPTIONS OF THE EXAMPLE

Table below lists the available plant measurements and their accuracy.

STREAM	DATA	VALUE	ACCURACY	PHYSICAL UNITS
AIR.1	MF O2	21	CST	mol%
AIR.1	MOLF	215000	5%	Nm3/h
AIR.1	P	1.013	2%	bar
AIR.1	T	15	1	C
AIR.2	P	20	2%	bar
AIR.2	T	500	2	C
NG	MOLF	6000	2%	Nm3/h
NG	P	20	2%	bar
NG	T	15	1	C
FUME.2	MF O2	14	2%	mol%
FUME.2	P	1.02	2%	bar
FUME.2	T	600	2	C
POWER	POWER	16.1	1%	MW

TABLE 4 - MEASUREMENTS OF THE EXAMPLE

14.2 PRELIMINARY WORK

THE THERMODS

In this example, we need three different THERMODS:

- Air
- Gas (natural gas)
- Fumes

These 3 THERMODS are defined by default in the file **empty.bls**. Therefore, no additional THERMOD or compound that needs to be defined.

CHEMICAL REACTIONS

In our example, chemical REACTIONS could be defined for the combustion. We do not need however to define them as the burner will be modeled by a BATVAL where atomic balances are considered (balance by elements: C, H, O, N).

SELECTING THE UNITS MODELS

Our model will be made of four UNITS:

NAME	TYPE	DESCRIPTION
C-1	COMVAL	compressor
R-1	BATVAL	combustion chamber
T-1	TURVAL	turbine
G-1	BBXVAL	black-box splitting the turbine power into a net power and the power needed for the compressor

TABLE 5 - UNITS OF THE EXAMPLE

14.3 DEFINING THE MODEL WITH VALIMODELLER

CREATING THE MODEL FILE

- open (**File** | **Open**) the file **Empty.bls**
- select the **File** | **Save As** command
- enter a new name for the model, for example **gt.bls**
- check that **Edit mode** is **On** to enable model modifications
- activate also **Options** | **Add Multiple Objects**

CREATING THE PFD

- select the **Edit** | **Add PFD** command to create a PFD
- click anywhere in the MAIN to indicate the position
- change the default name PFD1 by GT
- press the **OK** button

CREATING THE UNITS

Figure below shows where to place the UNITS

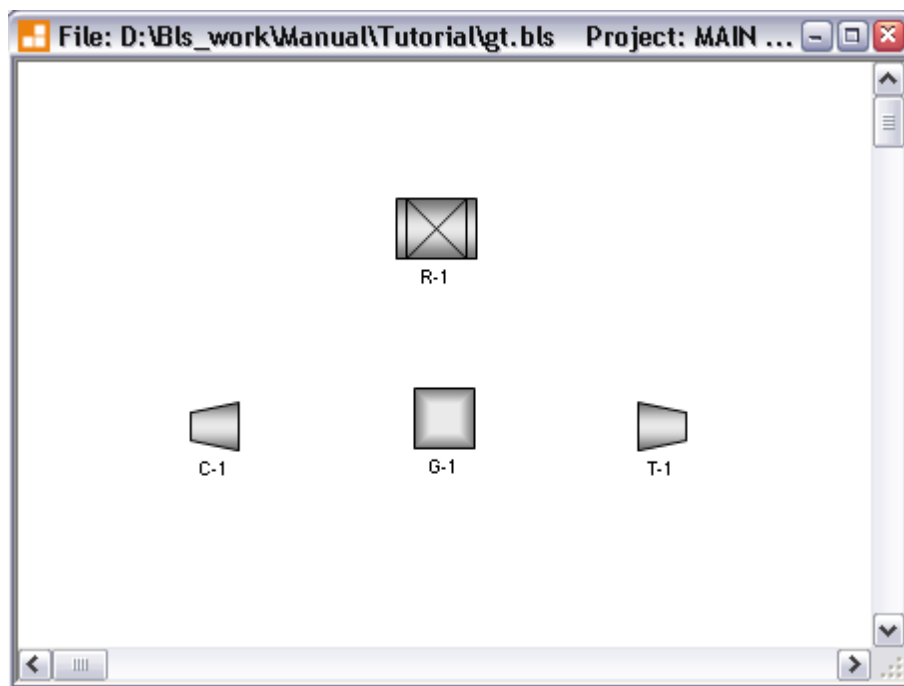


FIGURE 55 - DEFINING THE UNITS

COMPRESSOR C-1

- double-click the PFD GT to open it
- select the **Edit|Add Unit** command
- click in GT to specify the position of the Unit
- select the **Compressor COMVAL** Unit and press **OK**
- select symbol **Compressor Up/Down**
- check the Mirror Horizontal and Vertical boxes
- go into the **Tags** page
- click the **New** button
- select the EFFIC measurement in the right table
- activate the measurement with the  button
- change the measured value to 77 %
- click **OK** to accept changes in the Edit Window
- click **OK** to create the Tag
- click **OK** to create the Unit

COMBUSTION CHAMBER R-1

- click in **GT** to specify the position of the Unit R-1
- select the **Reactor BATVAL** Unit and press **OK**
- select symbol **Horizontal Reactor**
- select the pressure equation: **DP variable**
- go into the **Tags** page
- click the **New** button
- select the DP measurement in the right table
- activate the measurement with the  button
- change the measured value to 0.1 bar
- and the accuracy to 0.01 bar (Type A for absolute)
- click **OK** to accept changes in the Edit Window
- click **OK** to create the Tag
- click **OK** to create the Unit

TURBINE T-1

- click in **GT** to specify the position of the Unit
- select the **Turbine TURVAL** Unit and press **OK**
- select symbol **Turbine-Up/Down**
- check the Mirror Horizontal box
- go into the **Tags** page
- click the **New** button
- select the EFFIC measurement in the right table
- activate the measurement with the  button
- change the measured value to 80 %
- click **OK** to accept changes in the Edit Window
- click **OK** to create the Tag
- click **OK** to create the Unit

BLACK BOX G-1

- click in **GT** to specify the position of the Unit
- select the **Black box BBXVAL** Unit, change the Unit name by **G-1** and press **OK**
- select symbol **Black box**
- click **OK** to create the Unit
- press **ESC** to stop the creation of units

SAVE YOUR WORK

- select the **File | Save** command
- should the History window appears (depending on your configuration settings), press the **Cancel** button

CREATING THE STREAMS

Figure below will help you to create your Streams.

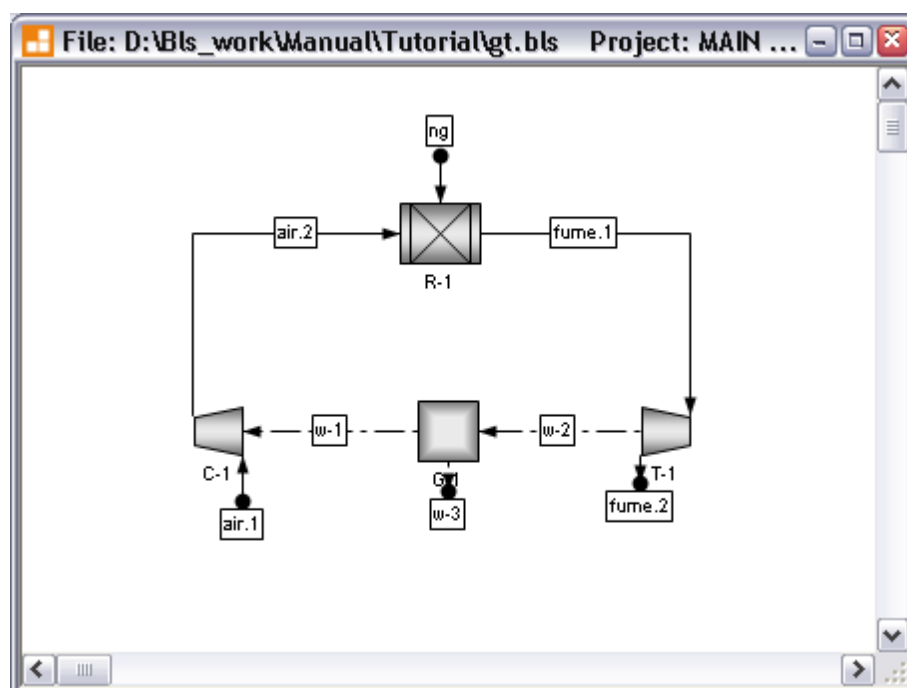


FIGURE 56 - DEFINING THE STREAMS

STREAM AIR.1

- select the **Edit | Add Stream** command
- select **Material** stream and Thermod **Air**
- change the name to AIR.1 and press OK
- press the Space Bar key on your keyboard to activate the input hotspots
- click on the inlet of the compressor C-1
- In the Stream notebook go to the Tags page
- click the **New** button
- select the T measurement in the right table
- activate the measurement with the  button
- measured value: 15 C, accuracy: 1 C (Type A for absolute)
- click **OK** to accept changes in the Edit Window
- select the P measurement in the right table
- activate the measurement with the  button
- measured value: 1.013 bar, accuracy: 2 % (Type % for relative)
- click **OK** to accept changes in the Edit Window
- select the MOLF measurement in the right table
- activate the measurement with the  button
- measured value: 215000 Nm3/h, accuracy: 10 % (Type % for relative)
- click **OK** to accept changes in the Edit Window
- select the MF measurement in the right table and the compound O2
- activate the measurement with the  button
- measured value: 21 % , accuracy: 0
- click **OK** to accept changes in the Edit Window
- click **OK** to create the Tags
- click **OK** to create the Stream

STREAM AIR.2

- select **Material** stream and Thermod **Air**
- change the name to AIR.2 and press OK
- click on the outlet of the compressor C-1
- click on an inlet of the reactor R-1
- In the Stream notebook go to the Tags page
- click the **New** button
- select the T measurement in the right table
- activate the measurement with the  button
- measured value: 500 C, accuracy: 2 C (Type A for absolute)
- click **OK** to accept changes in the Edit Window
- select the P measurement in the right table
- activate the measurement with the  button
- measured value: 20 bar, accuracy: 2 % (Type % for relative)
- click **OK** to accept changes in the Edit Window
- click **OK** to create the Tags
- click **OK** to create the Stream

STREAM NG

- select **Material** stream and Thermod GAS
- change the name to NG and press OK
- press the Space Bar key on your keyboard to activate the input hotspots
- click on an inlet of the reactor R-1
- in the Stream notebook go to the Tags page
- click the **New** button
- select the T measurement in the right table
- activate the measurement with the  button
- measured value: 15 C, accuracy: 1 C (Type A for absolute)
- click **OK** to accept changes in the Edit Window
- select the P measurement in the right table
- activate the measurement with the  button
- measured value: 20 bar, accuracy: 2 % (Type % for relative)
- click **OK** to accept changes in the Edit Window
- select the MOLF measurement in the right table
- activate the measurement with the  button
- measured value: 6000 Nm³/h, accuracy: 2 % (Type % for relative)
- click **OK** to accept changes in the Edit Window
- select the MF measurement in the right table and the compound C1
- activate the measurement with the  button
- measured value: 100 %, accuracy: 0 (Type % for relative)
- click **OK** to accept changes in the Edit Window
- click **OK** to create the Tags
- click **OK** to create the Stream

STREAM FUME.1

- select **Material** stream and Thermod **FUMES**
- change the name to FUME.1 and press OK
- click on the outlet of the reactor R-1
- click on the inlet of the turbine T-1
- click **OK** to create the Stream

STREAM FUME.2

- select **Material** stream and Thermod **FUMES**
- change the name to FUME.2 and press OK
- click on the outlet of the turbine T-1
- click again on the turbine
- in the Stream notebook go to the Tags page
- click the **New** button
- select the T measurement in the right table
- activate the measurement with the  button
- measured value: 600 C, accuracy: 2 C (Type A for absolute)
- click **OK** to accept changes in the Edit Window
- select the P measurement in the right table
- activate the measurement with the  button
- measured value: 1.02 bar, accuracy: 2 % (Type % for relative)
- click **OK** to accept changes in the Edit Window
- select the MF measurement in the right table and the compound 02
- activate the measurement with the  button
- measured value: 14 %, accuracy: 2 % (Type % for relative)
- click **OK** to accept changes in the Edit Window
- click **OK** to create the Tags
- click **OK** to create the Stream

STREAM W-1

- select **Mechanical** stream and press OK
- click on an outlet of the black box G-1
- click on the compressor C-1
- click **OK** to create the Stream

STREAM W-2

- select **Mechanical** stream and press OK
- click on the mechanical outlet of the turbine T-1
- click on an inlet of the black box G-1
- click **OK** to create the Stream

STREAM POWER

- select **Mechanical** stream and press OK
- click on an outlet of the black-box G-1
- click again on the black-box
- in the Stream notebook go to the Tags page
- click the **New** button
- select the POWER measurement in the right table
- activate the measurement with the  button
- measured value:16 100 kW, accuracy: 1 % (Type % for relative)
- click **OK** to accept changes in the Edit Window
- click **OK** to create the Tag
- click **OK** to create the Stream
- click **CANCEL** to stop creating new streams.

SAVE YOUR WORK

- select the **File | Save** command

You are now ready to run your model as shown in following chapter.



CHAPTER 5

HOW TO RUN YOUR DATA VALIDATION MODEL

SECTION 1: INTRODUCTION

In the previous chapter, you were introduced to ValiModeller GUI, which allows you to set-up and prepare the data validation model.

In this chapter, you will now learn how to run this model.

Section 2 **Running the model**

This section shows how to start ValiEngine from ValiModeller GUI and to review the results. This section also shows how to run ValiEngine independently from ValiModeller GUI.

Section 3 **Using TAG Files**

The TAG files are used to define the measurements by giving them a name, indicating to which variable they refer and defining their precision.

Section 4 **Using Measurement Files**

MEA files allows to exchange information between your VALI model and the outside words: getting on-line values of the measurements, changing starting point, setting OFF or OUT parts of a model, extracting results, writing reports, etc.

Section 5 **The Sensitivity Analysis**

Using the Sensitivity Analysis tool, you can identify how various measurements values and precisions influence each other.

This section also describes the mathematical basis of VALI and its compliance to the VDI-2048 guideline.

Section 6 **Tricks and Tips**

This section tackles some typical problems encountered while using VALI like how to initialise a model or how to solve convergence problems.

Section 7 **VALI Error Messages**

The error and information facility handling of VALI is described. Additional comments and recommendations are given for some of those messages.

SECTION 2: RUNNING THE MODEL

This section shows you how to run the model directly from the ValiModeller GUI or as a stand alone application. It also indicates how to browse VALI result files and how to load VALI results back in the interface.

2.1 STARTING VALIENGINE FROM VALIMODELLER GUI

To start VALI:

- select the **Application | Calculate** menu command or click  button
- fill-in the fields of the Specify runtime options dialogue box
- press the OK button
- then optionally fill-in the [Project history window](#)
- ValiEngine is then launched in a separate window (see below)

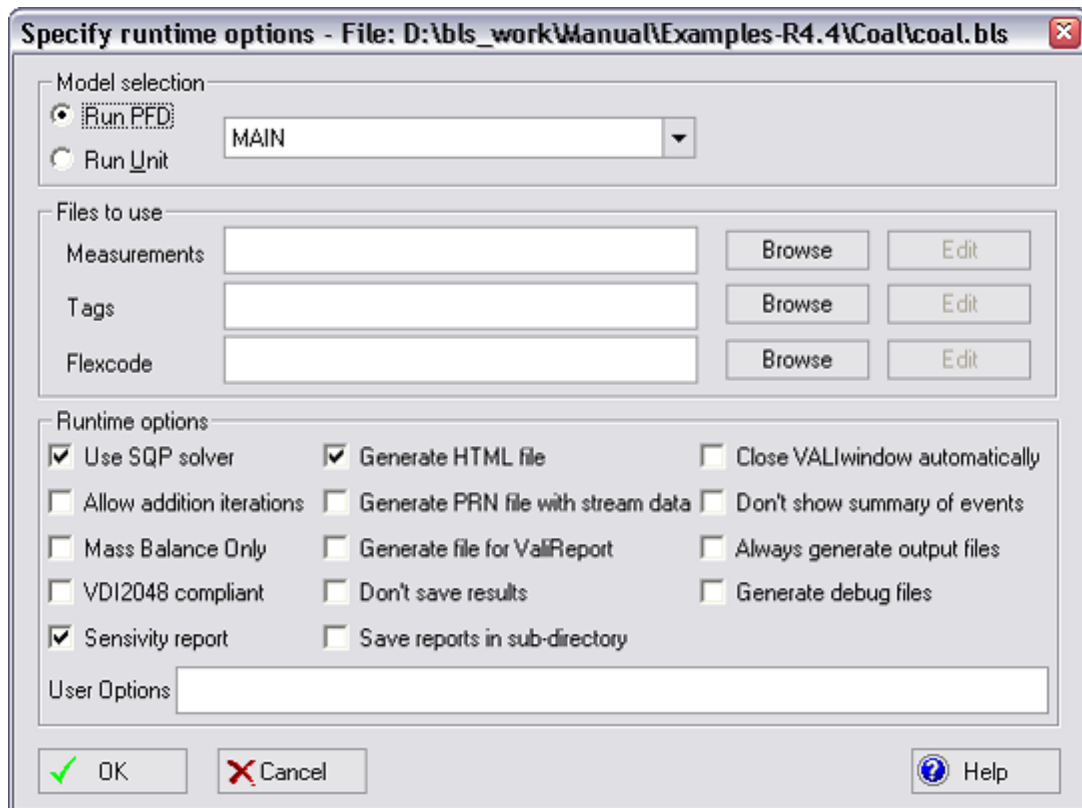


FIGURE 1 THE RUN VALI DIALOG BOX

- Model selection: Name of the PFD or UNIT defining the model.
- Measurements file: Name of the file containing the values of the measurements, as well as unit parameters, initial values, report specifications, etc. (optional)
- TAG file: Name of the TAG file (optional). When using a TAG file, the tags defined in the model are ignored
- Flexcode file: Name of the file containing flex code to be imported (optional)

A number of Runtime options can be specified through check boxes:

- Use SQP solver (-SQP) (activated by default): ValiEngine will invoke the SQP-IP solver.
- Allow additional iterations (-ITM) : ValEngine will ask for additional iterations if not converged after the number of iterations specified in the OPTVAL - ITMAX parameter.
- Mass Balance Only (-MBO): Only mass balances equations will be generated, for all units, whatever the user has selected for the equations (parameter MODE, for example). Additional equations created by OPXVAL units and Flex code will be kept.
- VDI2048 compliant (-VDI): This option activates the VDI compliant mode of VALI (use of 95% confidence level, penalty defined on correction accuracy, sensitivity analysis for all flagged variables, reports according to VDI 2048 guideline).
- Sensitivity report (-SNS): After the run, a Sensitivity Analysis report will be generated.
- Generate HTML file (-HTM) (activated by default): VALI will create HTML (Hyper Text Meta Language) formatted files for the reports. A number of active fields allow to navigate between the various sections of the reports, from streams to units, from and to TAG definitions, etc.
- Generate PRN file with stream data (-PRN): Vali will generate an Excel compatible file with stream data.
- Generate file for ValiReport (-ARC): VALI will produce a default ARC output file with the measured and validated value and accuracy and the comments of all tags as well as the date and time, the total penalty, the chi-square test value and sum of square of residues. If defined, the contents of RUNDATA and INFOi records will be printed as well. This file can directly be imported into ValiReport (see [Output and Archive Files](#) for more details about Archive (ARC) files).

- Don't save results (-NOS): The results will not be saved in the model file even if the run is successful (-NOS).
- Save reports in sub-directory (-SUB): The report files are generated in a sub-directory called results.
- Close VALI window automatically (-BAT): The VALI window will automatically be closed after the run. This option is useful for unattended processing in an online environment.
- Don't show summary of events (-R2V): VALI does not display any message on the screen, but writes them in the <model>.r2v file.
- Always generate output files (-NOC): VALI will generate output files specified in the mea file even if the run is not successful. This option has the same effect as placing the keyword NOCONVER in the mea file.
- Generate debug files (-DBG): Setting this option will generate three additional files when running the model, with a trace of the convergence process through the iterations:
 - r6v_itr.tmp: tags in decreasing order of penalty
 - resi_itr.tmp: equation residues by decreasing (absolute) value
 - vari_itr.tmp: values of the variables by decreasing relative change from preceding iteration

The same option followed by a unit name (-DBG unit) will generate a fourth file (unit.dbg) giving the derivatives of the equations units at each iteration.

User options

Additional options are:

- -ARA: same as -ARC but all the precisions are printed in absolute
- -COM: commentaries are commentaries only (without this option, some commentaries can be used during computation; eg tanks may use commentaries as product)
- -DIF: to print the difference (mea.val. - rec.val) for each tag in a dedicated file (<model>.dif)
- -HTO: create html reports with the lay-out of version 4.3
- -ITF: automatically register new tags appearing in the TAG file
- -JAC: print the Jacobian in the r3v file (requires the -R3V option to work)
- -NLV: skip the verification, made in the report phase, that L and V streams are actually in L or V state at the validated temperature and pressure
- -NOR: reduces r4v and r6v reports to simple run information, thereby accelerating the calculation
- -R3V: writes mathematical details in r3v file
- -SA2: the results will be saved as a separate model file named <model>_lastrun.bls
- -2SL: if the selected solver fails to converge, ValiEngine will automatically restart the calculations using the alternate solver
- -FLT: VALI stops after the verification phase. A specific report containing information about the filtering results is generated. If -ARC is used, a filtering ARC file is created as well

Attention: When starting ValiEngine from ValiModeller GUI, the file pc.vif is created with the settings you specify in the Run VALI dialogue box. The Runtime options appear as special keywords as indicated between brackets.

ValiEngine will go through the following steps:

- Phase 0: TAG reading
- Phase 1: Verification
- Phase 2: Analysis
- Phase 3: Resolution
- Phase 4: Report
- Phase 5: Output

If fatal errors are encountered during one of these phases, VALI will stop before starting the next step. You should then look for more information on the errors in the file <model>.r1v.

If the run is successful, then the results are saved in the model file (the -NOS runtime option disables this saving).

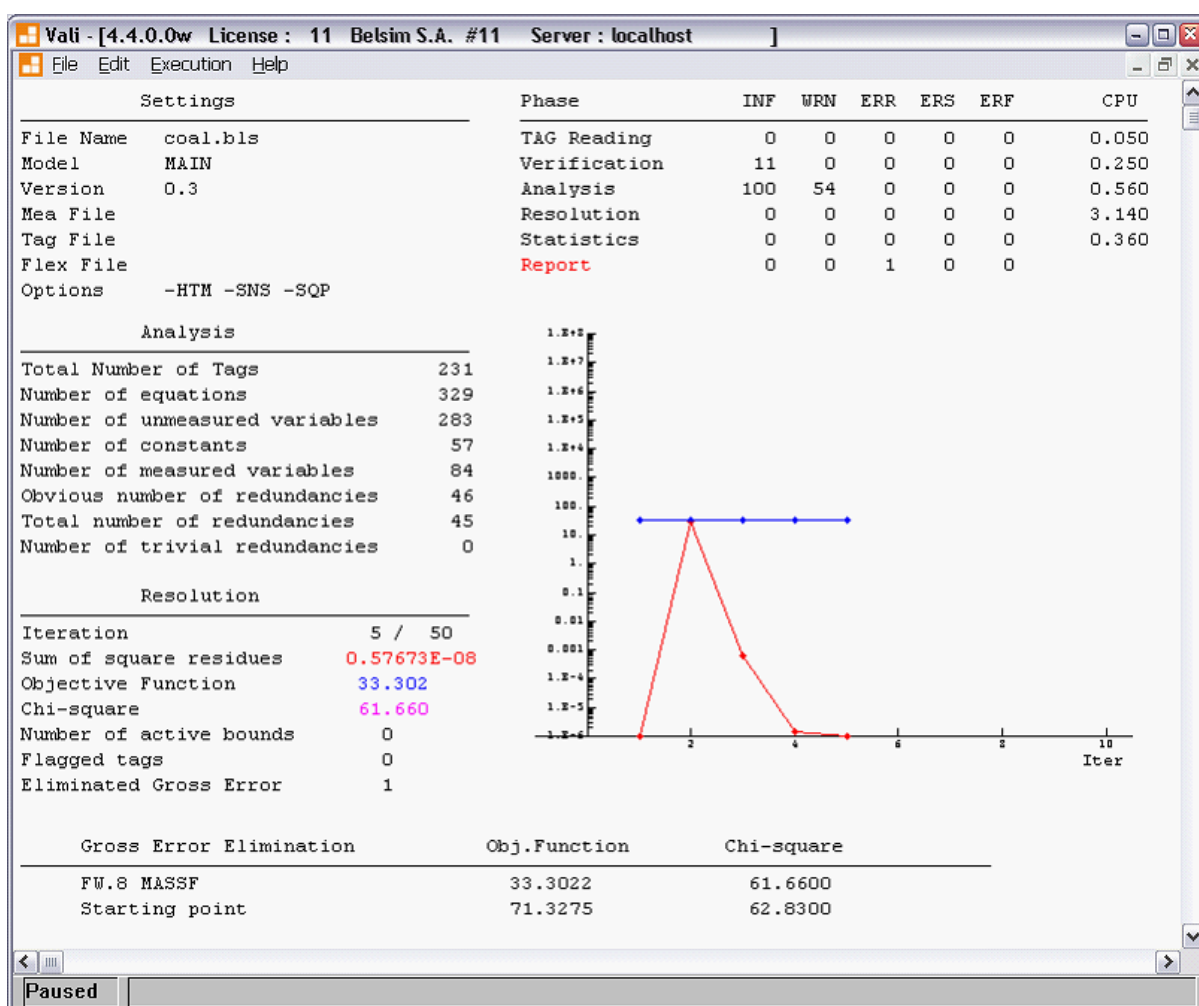


FIGURE 2 - VALIENGINE WINDOWS

2.2 INTERACTING WITH VALIENGINE

While running ValiEngine, four menus are available at the top of the window with submenu entries.

- File
 - Save As: saves the window content as a bitmap file at the location selected by the user
 - Print: prints the window content
 - Exit: closes the window and stops the execution after releasing the licence. After execution, you can also use the keyboard shortcut CTRL+C
- Edit
 - Select All: this allows selecting the content of the window. This action also results in Pausing the application. The execution will resume after copying in the clipboard (menu Copy). It can also be resumed through the command Execution|Resume
 - Copy: this menu item is greyed until activating Select. This command will copy the content of the window to the clipboard
- Execution
 - Abort: Closes the window of the application and stops the execution after releasing the licence. The difference with Exit is that a confirmation will be requested.
 - Pause: Pauses the application. When the programme is paused, this menu item is toggled to Resume. You can also use the keyboard shortcut CTRL+S to pause and CTRL+Q to resume the execution.
 - Stop: Stops the application and generates reports available at that stage when the current calculations will be finished. It is possible to invoke this during the first three phases or between two iterations.
 - More Iterations: Activates/deactivates the option to have more iterations during resolution. This is equivalent to setting or not the -ITM option. A check mark is showing the state of the option.
 - Try Other Solver: Activates/deactivates the option to switch to the other solver if resolution fails. This is equivalent to setting or not the -2SL option A check mark is showing the state of the option.
 - Sensitivity Analysis: Activates/deactivates the option to run the sensitivity analysis after resolution. This is equivalent to setting or not the -SNS option. A check mark is showing the state of the option.
 - Save Results: Activates/deactivates the option to save the solution found by solver in the BLS file after resolution. This is equivalent to setting or not the -NOS option. A check mark is showing the state of the option.
 - File for ValiReport: Activates/deactivates the option to have more iterations during resolution. This is equivalent to setting or not the -ARC option. A check mark is showing the state of the option.

- Close Window Automatically: Activates/deactivates the option to close automatically the window after resolution. This is equivalent to setting or not the -BAT option. A check mark is showing the state of the option.
- Help
 - About: Opens an information window with the version of the ValiEngine
 - Help: Invokes the Help facility

Attention: The menu Execution is grayed after completion of the run.

2.3 BROWSING RESULT FILES

You can browse VALI result files directly from the interface:

- select the **Application | Results** menu command
- select the report you want to see
- the corresponding file is then loaded in your web browser (for html reports) or text editor (by default WordPad)

Alternatively you can access directly access the result files from Windows explorer.

VALI creates several files, whose names are derived by adding an extension to the model name (The extension htm is added to the file names when generating html files). These files can be created in a sub-directory called **results** through runtime option -SUB.

<MODEL>.HTM: GENERAL INFORMATION (HOME)

This file is only created when the HTM option is checked. It gives a first overview of the results. The links on the top (Home, Summary, etc.) give access to the different reports, while the links on the left side give a direct access to the different sections of the reports. The greyed sections are not available, according to the reporting options selected through unit [REPVAL](#).

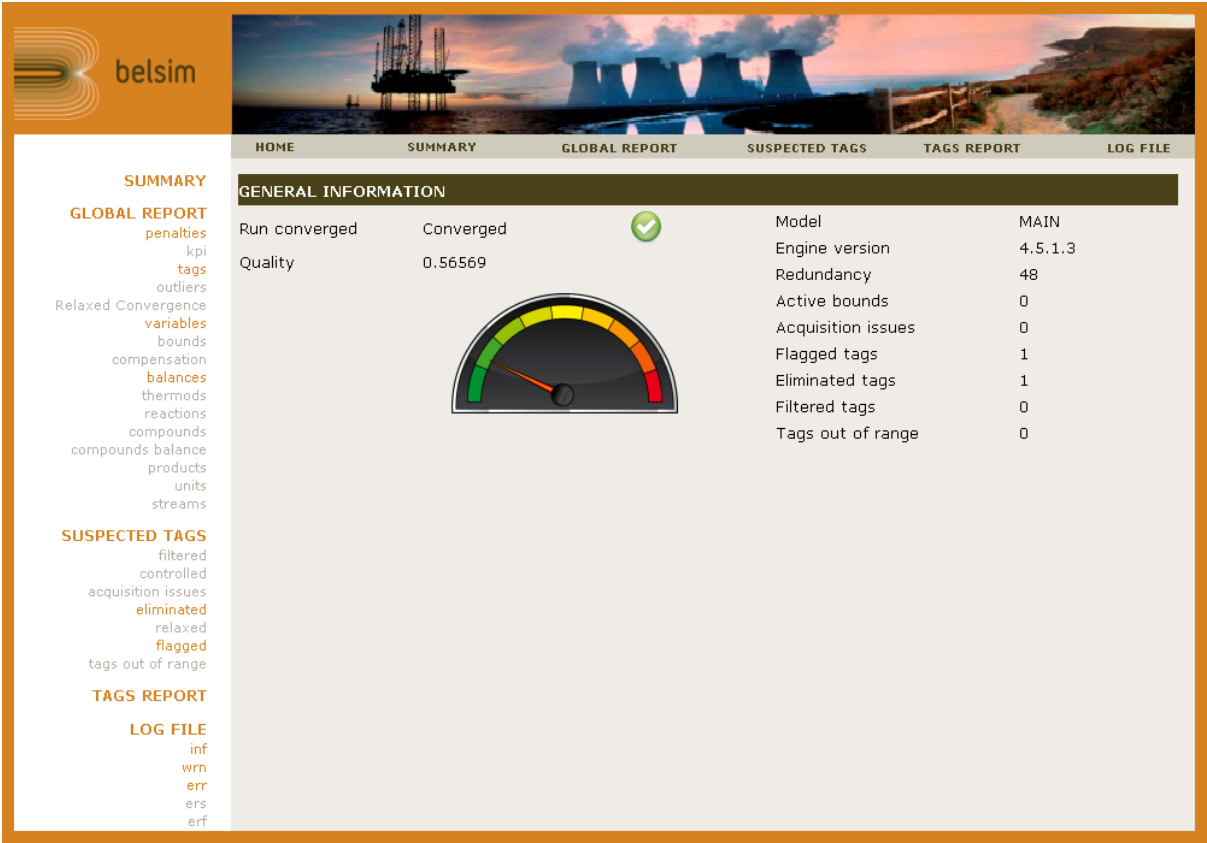


FIGURE 3 - GENERAL INFORMATION (HOME PAGE)

<MODEL>.SUM.HTM: SUMMARY

This file is only created when the HTM option is checked. It replicates most of the information provided in ValiEngine windows during the run.

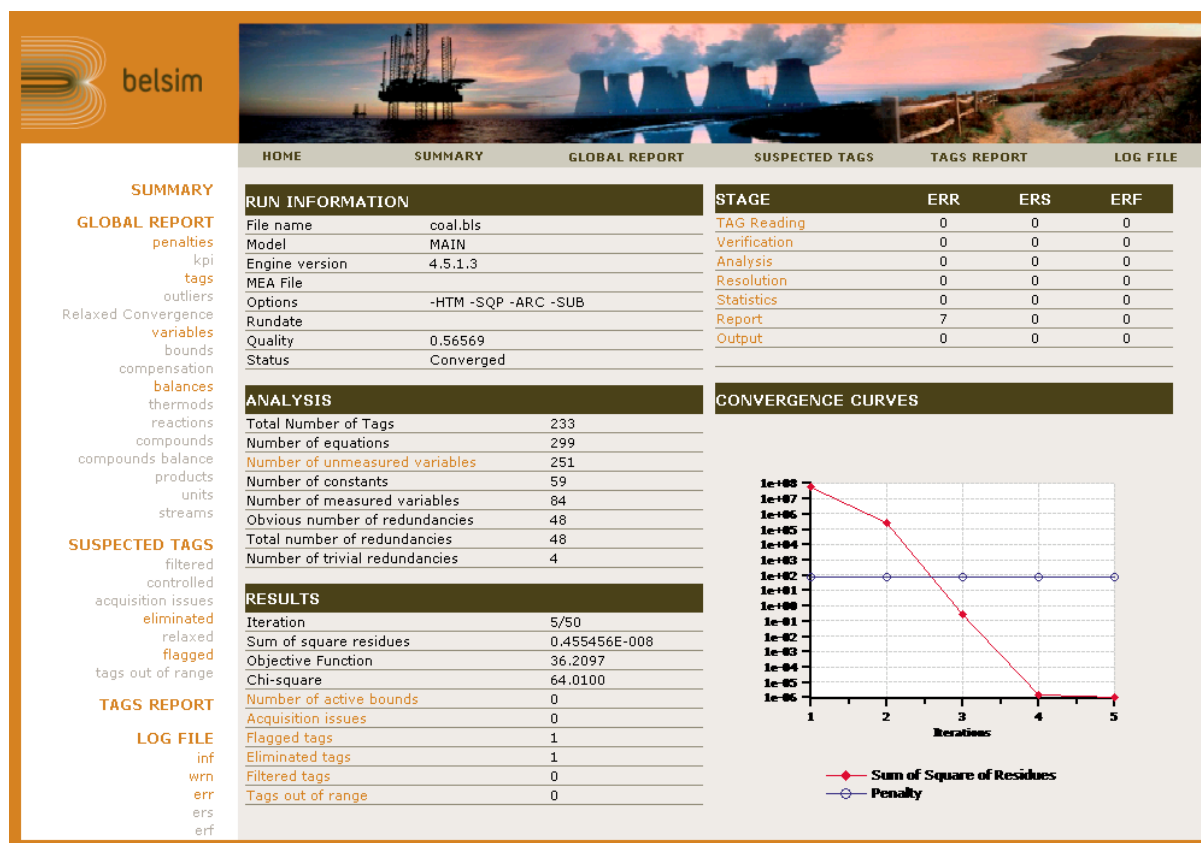


FIGURE 4 - SUMMARY

<MODEL>.R1V.HTM: LOG FILE

Summary of the run with various warning and error messages from the various phases.

<MODEL>.R2V.HTM: RUNTIME MESSAGES

This file is created only when running with **-R2S** option (Don't show summary of events) and contains the messages that are usually sent to the screen.

<MODEL>.R3V.HTM: MATHEMATICAL DETAILS

This file is created only when running with **-R3V** option and contains the list of equations and variables as well as a trace of the convergence algorithm.

<MODEL>.R4V.HTM: GLOBAL REPORT

This is the main report file. Its contents can be customized through the unit REPVAL. It contains for example variables report, units report, tags report, etc.

<MODEL>.R5V.HTM: SUSPECTED TAGS

List of measurements that are suspected, i.e. tags for which one of the following problem has been detected:

- acquisition issue: tags that was not acquired correctly (read several times, not found in mea file, read as BAD)
- filtered
- eliminated ,flagged or relaxed as suspicious because their correction is bigger than expected
- out of range: reconciled value is outside the filtering bounds

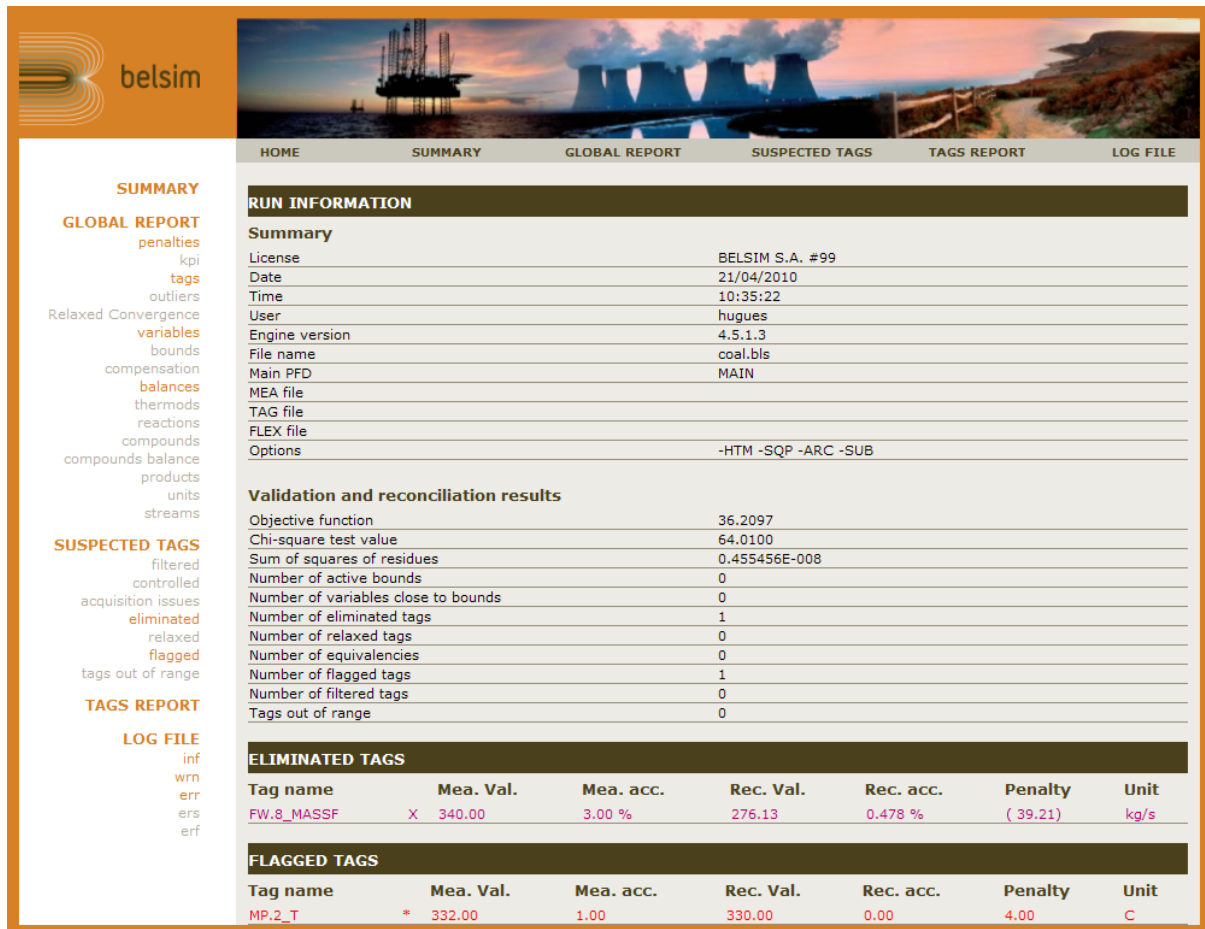


FIGURE 5 -SUSPECTED TAGS

<MODEL>.R6V.HTM: TAGS REPORT

This file contains information about all the measurements in the model, i.e.:

- number of equations, variables, etc.
- total penalty
- list of measurements with their measured and validated values and accuracies

After a successful run, VALI will also create a back-up file called from the model and PFD name as **<model>_<PFD>.bls.bak**. This file is a copy of the model file as it was when VALI was started. This gives you the opportunity of re-importing the last running version of a model, by removing the **bak** extension from the backup file name.

Attention: When running in batch mode, you can specify your own name for the back-up file by changing line 7 of the **pc.vif** file.

<MODEL>.FLT.HTM: FILTERING REPORT

This file is only generated when the option **-FLT** is used. It contains information about the filtering phase, i.e.:

- list of acquisition issues
- number of acquisition issues by PFD
- list of filtered tags
- list of objects set OFF and OUT in measurement files
- list of objects set OFF by cutoff
- list of objects set OFF by FLEX filtering
- table of comparison between default and measured value
- sequences of propagation of OFF status

2.4 LOADING VALI RESULTS BACK IN VALIMODELLER GUI

After running VALI (or when any other application has changed the model file), ValiModeller GUI will propose:

- to **reload** the results: only the data are read from the file, assuming the model structure has not changed. This operation might fail if some structural changes have been made to the model by the external application
- to take **no action**: the results are not loaded and the next run will restart from the initial point presented by ValiModeller GUI

The latter option allows you to first look at the results before loading them. Depending on these results you can decide later on to load the results back in ValiModeller GUI and use the calculated results as a new initial point for the next run, through the command **Application | Load Results** (or

its corresponding button ,).

Attention:

1. When running VALI, the results are saved in the file only if the run is successful. Only the measured values coming from measurement files will be saved in the model file if the run is not successful.

2. Even if the run is successful, the runtime option -NOS allows you to suppress any saving of the results by VALI in the model file.
3. In these two cases, the Application | Load results command will restore the starting point as it was before the last run.

2.5 USING VALI IN BATCH MODE

STARTING VALI

Provided you have a ValiOnlineEngine license, you can start the VALI.EXE application as any other Windows applications:

- by defining a short-cut to VALI.EXE on your desktop (or elsewhere) and clicking on it. Do not forget to indicate in the properties of the short-cut in which directory it must start
- by associating VALI.EXE to all files of type "vif" and clicking on any of these files (see below for the contents of these files). It is also recommended to define an "edit" option on the vif files. This allows to open them in a text editor (right click|edit) for modifications or to start Vali.exe (double-click)
- by opening a DOS Command windows, going in the appropriate directory and entering the command: [InstallDir] \bin\vali.exe [file] where the optional file keyword is the name of the vif file described below (by default pc.vif) and [InstallDir] stands for the name of the folder where Vali has been installed

Attention:

1. You can also include the start-up command in a DOS batch file.
2. Running VALI in an automated manner (meaning without human intervention) requires a ValiOnlineEngine, which is made available separately from the ValiModeller.
3. You can also run the VALIAUTO.exe application, which is strictly equivalent to VALI.EXE except it does not produce any output on the screen.

USING THE "LASTRUN" MODEL FILE

As any equation solver based tool, Vali convergence properties may depend on the initial point. When running online (e.g. every hour), the best initial point is usually the validated data of the preceding run as the operating conditions usually do not change drastically from run to run. In some cases however, like in start-up conditions after a shut-down, starting from the last run data might not be appropriate.

The notion of **Lastrun** model file has been developed to maximize convergence rate using the following strategy:

- the calculations are first started with the model as saved after the last successful run
- if the model fails to converge, then the calculations are restarted on basis of a reference model, saved with typical operating conditions data

This strategy is launched through the combined use of the runtime options -NOS (which disables results writing in the reference model file) and -SA2 (which saves the converged results in a separate file - the so-called Lastrun model).

The Lastrun model file is named after the name of the reference model, followed by the suffix **_lastrun**. If you use a separate folder for the results (-SUB) then the lastrun file will be created there as well.

VALI INFORMATION FILE

When starting, VALI needs to know the model to be run as well as the TAG and MEA files to be used.

This information is normally defined in a Vali Information File (default extension **vif**). If not given as argument, VALI will look by default for a file named **pc.vif**. If this file is not present in the current directory, the information will be requested interactively.

line 1: name of the model file

line 2: name of the model (PFD)

line 3: runtime options

line 4: name of the Flex file

line 5: name of the MEA file

line 6: name of the TAG file

line 7: name of the model back-up file

line 8: name of second MEA file

line 9: name of third MEA file

Example:

```
gt.bls
GT
-htm
GT.MEA
GT.TAG
```

RUN STATUS (ERRORLEVEL)

At the end of a run, VALI indicates as an exit code the run status. This status can be checked through the DOS command "IF ERRORLEVEL". The following codes are used:

- -3: System error - no connection to the license server or licence issue
- -2: System error - installation issue: missing directory, wrong version number
- -1: Fatal error
- 0: The run was successful
- 1: The run was successful but gross errors are suspected to be present (according to Chi-square test)

A file named **valiexit.log** is also created with one line containing the keyword EXITCODE followed by the same code. (Example: EXITCODE 0)

SECTION 3: USING TAG FILES

TAG files allow to define the TAGS (i.e. the measurements) to be taken into account for your model.

TAGS can be defined either inside the model with ValiModeller GUI or in a separate file called a TAG file. When using a separate TAG file, the TAGS defined in the model (if any) will simply be ignored.

Attention: Using the command **File|Import|TAG file** you can import into your model TAGs defined in a TAG file. After this command you must re-open the file.

A TAG file contains the definitions of your TAGS (or measurements): tag name, associated process variable, default measurement value and accuracy, default physical units and control strings.

Several TAG files can be used to define your measurements. The main TAG file must then include (by using the INPUT keyword) the other TAG files. Control strings are optional and are used:

- to issue various messages depending on the measured and validated values of measurements
- to change the accuracy of measurements depending on their range
- to provide default value when measurements are not available or out of range

3.1 GENERAL SYNTAX OF A TAG FILE

All keywords should be in Uppercase, except comments and messages.

NUMBERS

The decimal point can be either "." or ",".

COMMENT SYMBOL "!"

Blank lines and anything appearing on a line after the comment symbol "!" are ignored. Each TAG definition starts on a new line.

CONTINUATION CHARACTER "&"

Each TAG definition starts on a new line. When a TAG definition does not fit on one line, the continuation symbol "&" must be used at the end of the line that must be continued.

INPUT FILES

The INPUT command may be used to include TAG definitions from within other TAG files. INPUTs may be nested up to 15 levels.

Example:

```
INPUT list2.tag ! this includes the contents of list2.tag in the current file.
```

3.2 TAG DEFINITION

The syntax for a TAG definition is:

```
TAG    TYP    NAM    VAR    VAL    ACC    PU    ["STR1" ["STR2" ["STR3"]]]
```

Where:

- TAG: the TAG name (the name of the sensor) limited to 32 characters
- TYP: the type of the object (STREAM or UNIT) that owns the measured variable
- NAM: is the name of the object (STREAM or UNIT) that owns the measured variable
- VAR: is the name of the measured variable:
 - UNIT:
 - DPi: pressure drop
 - A: area of a HTXVAL unit
 - EFFIC: efficiency of a turbine, compressor or pump
 - any measurable parameter of a UNIT
 - Material STREAM:

- T: temperature
- P: pressure
- H: enthalpy:
- any measurable parameter of a STREAM
- Thermal STREAM:
 - LOAD: heat load
- Mechanical STREAM:
 - POWER: mechanical power
 - RPM: rotations per minute
- VAL: is the default value for the measurement
- ACC: is the default accuracy for the measurement
- PU: are the default physical units for the measurement
- STR1: is the optional control string for messages and must be enclosed between double quotes
- STR2: is the optional control string for measured value and must be enclosed between double quotes
- STR3: is the optional control string for accuracy and must be enclosed between double quotes

3.3 CONTROL STRINGS

The Control Strings are used to carry out various actions depending on the measured and sometimes validated value of a given TAG: issuing messages, replacing the measured value by another value or changing its accuracy.

The syntax is described in chapter "[How to build your data validation model](#)".

Please note that you must not include the &1 &2 and &3 characters. The string type is detected from its position in the list of strings.

3.4 DEFINING DUMMY TAGS

You can also define dummy TAGS which will be ignored by VALI. This is useful when the values of your measurements are coming from a process control computer and contain also values of TAGS that are not relevant for the data validation model. Defining these TAGS as dummy will avoid getting numerous VALI warning messages about undefined TAGS.

The syntax is:

NAME **DUMMY**

where:

- NAME: the tag name

Example:

T101 DUMMY

3.5 EXAMPLE OF A TAG FILE

```
T101    STREAM S101    T            10        5        C
P101    STREAM S101    P            3.5      0.1      bar
T117    STREAM S117    T            25        5        C &
"'VERY LOW' (<15), 'LOW' (15,20), 'HIGH' (30,35), 'VERY HIGH' (>35)" &
"20 (<20), 30 (>30)" &
"OFF (<15), 50 (15,20), 50 (30,35), OFF (>35)"
P117    STREAM P117    P            2           0.1      bar
INPUT list2.tag
INPUT list3.tag
```

SECTION 4: USING MEASUREMENTS FILES

Measurement files are used to exchange information between VALI and the outside world . They are primarily used to enter the actual values of the measurements. They can also be used to modify various model parameters, to enter initial values of UNITS and STREAMS or to set them OFF or OUT. Finally, measurements files can also be used to produce result files with validated values for reporting purpose.

The Measurement file facility is a very flexible and powerful way of exchanging information between VALI and the outside world: yourself as a user, but also process control computer, LIMS, archiving systems, etc. Based on ASCII files of simple structure, this system allows easy cross-platform communication with various hardware systems.

A measurements file can contain:

- actual measurement values, coming for example from the plant or from the lab
- descriptive information about the run
- updated values of tags: accuracies, control strings, bounds
- settings of ValiFilter
- unit parameters
- initial values of measured or unmeasured variables
- lists of STREAMS and/or UNITS to be set OFF
- lists of UNITS to be set OUT
- compound data
- values of the binary parameters of THERMODS
- tags to be generated in the output files
- variables to be written in the output files
- report specifications

4.1 SYNTAX OF A MEASUREMENTS FILE

Each command in a measurements file starts on a new line. All keywords should be in Uppercase, except comments and strings.

NUMBERS

The decimal point can be either "." or ",".

COMMENT SYMBOL "!"

Blank lines and anything appearing on a line after the comment symbol "!" are ignored.

INPUT FILES

The INPUT command may be used to include TAG definitions from within other TAG files. INPUTs may be nested up to 15 levels.

Example:

```
INPUT list2.mea ! this includes the contents of list2.tag in the current file.
```

Attention: when creating OUTPUT and ARCHIVE files, the use of INPUT files is restricted.

4.1.1 STRAP AND TABINPUT FILES

The STRAP and TABINPUT files are used to include tabular data related to model units, like strapping tables of storage tanks (TNKVAL units) and trajectories of wells and pipes (PIPEVAL unit). STRAP and TABINPUT keywords are synonyms and can be used indifferently for any table.

Example:

```
TABINPUT L01-WELL.txt ! imports the trajectory defined in L01-WELL.txt
```

The tabular data must be entered with the following syntax (keywords are in **bold**):

```
UNITNAME      L01-WELL      ! Name of a PIPEVAL/TNKVAL unit
NUMTABLE      1            ! Table ID (currently only 1 table is supported on a unit)
MAXCOL 2      ! Number of columns of the table
PHYSUN LG m   ! Physical units definition for each column
300.0 -2349.5      ! Tabulated values
186.0 -2463.4
43.0  -2606.3
0.0   -2649.3
```

Attention: Physical units definition include the type of units (see a complete list of units types in section User-defined Physical UNITS in this Chapter) as well as the engineering units that are used for the data. In the above example, the keywords LG indicate that the data correspond to a Length.

4.1.2 DICTIONARY FILES

Dictionary files define or overwrite default definitions of various VALI keywords dealing with:

- Status of active and inactive streams (STATUS tags)
- Products codes and IDs
- Movement types
- Sub-Movement types
- Suffix of tags to be written back to historian

Dictionary definitions are given in one or several separate text file that are referred to in your measurement files through the following syntax:

DICTIONARY somefile.mea ! imports the definitions listed in somefile.mea

In the dictionary file itself, the keywords used with STATUS tags to indicate an active or inactive status are listed line by line after the keywords OPEN or CLOSED.

Example:

```
OPEN    "Open"
OPEN    "On"
CLOSED  "Closed"
CLOSED  "Off"
CLOSED  "Shutdown"
```

Product codes and IDs are defined through PRODUCT syntax.

Example:

```
PRODUCT CRUDE    101
PRODUCT KERO     102
PRODUCT LGO      103
```

For movement types, names to be used and corresponding tag names are defined through the following keywords:

KEYWORD	DEFAULT NAME	DESCRIPTION
TANK_CHARGE	CHARGE	Amount loaded to tank
TANK_RUNDOWN	RUNDOWN	Amount unloaded from tank
PRODUCTION	PRODUCT	Amount produced
CONSUMPTION	CONSUMP	Amount consumed
IMBALANCE	IMBAL	Product imbalance
OPENING_INVENTORY	OPINVENT	Opening inventory
CLOSING_INVENTORY	CLINVENT	Closing inventory
TRANSFER_IN	TRAN-IN	Amount transferred into the zone
TRANSFER_OUT	TRAN-OUT	Amount transferred out of the zone
IMPORT	IMPORT	Amount imported (receipts)
EXPORT	EXPORT	Amount exported (shipments)
TRANSFER	TRANSFER	Amount transferred out of ZoneFrom and transferred into ZoneTo

Example:

```
TANK_CHARGE "TANK-IN"
```

When necessary, the Sub-Movements will be defined: SHIPMENT, LOSSES, EXPORT...

Example:

```
MOVEMENT      STREAM Stream0001    EXPORT
SUBMOVEMENT    STREAM Stream0001    LOSSES
```

If no sub-movement is specified, the default movement name will be used as sub-movement name. In that case, the sub-movement name will be the default movement name (even if the movement name is changed by the dictionary, see further).

When generating a file with reconciled values of tags to be written back to an historian, based on the settings SUFFIX_V in the historian page of the tags, the actual suffix that will be used can be set through SUFFIX_V syntax.

Example:

```
SUFFIX_V "VALI"
```

4.1.3 INTERNAL VARIABLES AND IF THEN ELSE

Internal variables can be set using the syntax:

```
@SET VAR VALUE
@SET VAR "STRING"
@SET VAR TAGNAME
@SET VAR = VALUE OPERATION VALUE
@SET VAR = VAR1 OPERATION VAR2
```

Where:

- VAR: the internal variable name
- value: a numeric value (integer or real)
- string: an alphanumeric value
- TAGNAME: the name of a tag. (**Attention: The value of the tag will be expressed in Belsim internal units: K, bar, kg/s, kmol/s, etc.**)
- VARi: a previously defined variable name
- Operation: one of the 4 arithmetic operations (+ or - or / or x) (as * is considered as a joker, x is the multiplication sign)

These internal variables can then be used in the following commands:

```
@IF ( CONDITION ) THEN
SOME COMMANDS
@ELSE
SOME OTHER COMMANDS
@ENDIF
```

where **condition** can be:

```
VAR = "STRING"
VAR =, <,<=, >,>=  VALUE
VAR =, <,<=, >,>=  TAGNAME
```

Where:

- VAR: the name of an internal variable.

Attention:

1. The parentheses and the spaces around them are mandatory.
2. The ELSE clause is optional.
3. IF THEN ELSE clauses can be nested up to 10 levels.

Examples:

```
! Activates or deactivates some stream and uses a secondary file according to the
! operation mode
!
@SET SEC1 "ACTIVE"

@IF ( SEC1 = "ACTIVE" ) THEN
    STREAM Sec1_IN ON
    Input section1.mea
@ELSE
    STREAM Sec1_IN OFF
@ENDIF
!
! Shuts-down heat exchanger E-1 if outlet temperature below 100 C.
!
@SET T1 TI-001

@IF ( T1 < 373 ) THEN
UNIT E-1 OFF
@ENDIF
!
! Assumes stream s1 is active only if DT > 0.5 C
!
@SET DT = TI001 - TI002

@IF ( DT > 0.5 ) THEN
    STREAM s1 ON
@ELSE
    STREAM s2 OFF
@ENDIF
```

4.1.4 TEMPORARY SECTION

Sections of measurement files that will only be used during a specific time period can be defined using the syntax:

```
BEGIN_TEMPORARY "BEGINDATE" "ENDDATE"
TEMPORARY COMMAND
END_TEMPORARY
```

BEGINDATE and ENDDATE define the time period for which the temporary command is active. If

RUNDATE (TAG used to store the date of the run) is comprised between BEGINDATE and ENDDATE, the temporary command lines will be executed.

Example:

```
BEGIN_TEMPORARY "2010-01-01 00:00:00" "2011-01-01 00:00:00"
AIR.1_MOLF 220000 Nm3/h
AIR.1_T OFF
INPUT TempCommand.mea
END_TEMPORARY
```

Attention: Dates (BEGINDATE, ENDDATE and RUNDATE) must be expressed in the following format “yyyy-MM-dd HH:mm:ss”

4.2 CHANGING VALUES OF TAGS

Measurements files (or MEA files) are a simple way to change the measured values of the TAGS defined in the model or through a TAG file. Most of the modern Process Control Systems and DAHS can automatically generate this type of file with current values of the plant measurements. Lab data can also be sent to VALI via this kind of files.

Attention: ValiLink enables to automatically create these files by interacting with various information systems and/or databases.

The following syntax can be used:

TAGNAME	VALUE	[ACC]	[PU]
TAGNAME	VALUE	CST	[PU]
TAGNAME	[VALUE]	OFF	[PU]
TAGNAME	[VALUE]	BAD	[PU]

where :

- TAGNAME: is the TAG name
- VALUE: is the measurement value
- ACC: is the accuracy (optional)
- PU: are the physical units (optional)
- BAD: means no value is currently available for that tag. If a filter is defined on the corresponding tag, it will be activated. Otherwise, the default tag value will be taken in place of the measured one

The default values for **ACC** and **PU** are the default values defined in the model file (or the TAG file) for the corresponding TAG.

Attention:

1. TAGS for which no value are given in the mea files will keep their default value (as introduced in the TAG definition).
2. The default values of the TAG in the model file remains unchanged.
3. For STATUS tags, the value can be 1 (or OPEN) or 0 (or CLOSED), or any string value (enclosed between single quotes) defined through DICTIONARY syntax.

You can also change the failure probability through the syntax:

```
TAGNAME      FAILPROB      VALUE      [PU]
```

Another syntax can also be used to change properties (default value, default accuracy, bounds, failure probability and control strings) of several tags in one shot. In this case the tag values in the model file will be updated.

The following syntax can be used:

```
UPDATETAG    TYPE    SOMETYPE    APROPERTY    VALUE    [PU]
UPDATETAG    NAME    SOMENAME    APROPERTY    VALUE    [PU]
UPDATETAG    TYPE    SOMETYPE    CONTROL_STRINGS    [PU]
UPDATETAG    NAME    SOMENAME    CONTROL_STRINGS    [PU]
```

where:

- SOMETYPE and SOMENAME are strings defining which tags are to be changed (based on their type or name) and can contain wildcards
- APROPERTY is the name of the property to be changed (MEA.VAL, MEA.ACC, DEF.VAL, DEF.ACC, RAW.VAL, RAW.ACC, SL, LL, L, H, HH, SH, FAILPROB, COMMENT, IS.BAD, IS.FLAT1, IS.FLAT2)
- VALUE is the new value or **OFF** or **CST**
- PU is optional physical units,
- CONTROL_STRINGS are the control strings of the tags (a series of 3 strings enclosed by double quotes)

Attention: see DEFINING FILTER SETTINGS paragraph for a variant of UPDATETAG syntax enabling to change ValiFilter settings.

The third syntax enables to change the value of the tag. The tag will not be considered as filtered if a valid value is given.

The following syntax can be used:

```
FILTERED     TAGNAME     VALUE [ACC] [PU]
FILTERED     TAGNAME     [VALUE]    BAD    [PU]
```

where:

- TAGNAME: is the TAG name
- VALUE: the filtered measurement value
- ACC: the filtered accuracy (optional)
- PU: are the physical units (optional)
- BAD: means no value is currently available for that tag. If a filter is defined on the corresponding tag, it will be activated. Otherwise, the default tag value will be taken in place of the measured one.

You can also change the PLANTNAME of a tag using the following syntax:

TAGNAME **PLANT** **PLANTNAME**

Examples:

```
!
! Defines some measured values (no change of default values in the model file)
!
PI-001 20      bar
PI-002 34.23  -3      kg/cm2
PI-003 OFF
PI-004 21.23  CST
!
! Sets the default precision of all pressure measurements at 3%
! (changed in the model file)
!
UPDATETAG      TYPE    P      MEA.ACC      -3
!
! Changes the "measurement" control string (defining a CUTOFF value) of some tags
! (changed in the model file)
!
UPDATETAG      NAME    01*FF* " "  "CUTOFF(<0.1)"  " "
!
! Filters the value of tag PI-005
!
FILTERED        PI-005 2      bar
```

4.3 DESCRIPTIVE INFORMATION ABOUT THE RUN

A number of pseudo-tags are also available to store some information (80 characters each) which can then be reprinted in the output files and in ValiReport:

- **RUNDATE**: to store the date of the run (free format)
- **INFOi** (i=1..9): to store some other text

4.4 SETTING THE REFERENCE ATMOSPHERIC PRESSURE

Pressure measurements can be specified in absolute (bar, atm, Pa, etc) or relative - to atmospheric pressure - basis (barg, MPag, etc.). The reference atmospheric pressure is 1.0 bar by default. This value can be overwritten by including the following syntax in a measurement file:

ATMOPRES **VALUE**

where:

- VALUE: the current atmospheric pressure expressed in bar (absolute)

The atmospheric pressure can also be set through a tag defined on a [ATMOPRES](#) unit.

4.5 DEFINING BOUNDS ON TAGS

The handling of solver bounds only works with the new SQP solver. Do not forget to use the -SQP Vali runtime option.

Setting bounds of variables is done via the bounds specified on the TAGS. These bounds can also be set through measurements files using the following syntax:

```
TAGNAME [SL VALUE] [LL VALUE] [L VALUE] [H VALUE] [HH VALUE] [SH VALUE] [PU]
```

where:

- TAGNAME: the TAG name
- VALUE: the bound value
- PU: the physical unit

Examples:

```
TI-001 SL 200 C ! The variable defined by TI-001 has a lower bound of 200C
TI-002 SL 150 SH 400 C ! The variable defined by TI-002 must fall in the range 150-400 C
```

Attention:

1. The tags on which you set the bounds can be OFF. This allows to specify bounds on unmeasured variables as well.
2. To introduce inequality constraints between two variables (e.g. TI-001 > TI-002), you must use a unit of type OPXVAL to calculate the difference TI-001 - TI-002 and then set a bound on this difference. Alternatively you can also a FLXVAL unit.

4.6 DEFINING BOUND CONTROL

Attention: Bound Control settings are usually defined through ValiModeller GUI.

Bound control mechanism keeps selected variables (defined through a TAG) at a specified distance from their solver lower and/or upper bound.

The following syntax can be used:

```
TAGNAME BND_CONTROL ACTION CRITERIA [PU]
```

where:

- **ACTION:** the bound control mechanism:
 - **NONE:** no action is taken
 - **SL:** bound control on the lower solver bound
 - **SH:** bound control on the upper solver bound
 - **BOTH:** bound control on the both solver bounds
 - **Value Physun:** the measured value is replaced by the given value
- **CRITERIA:** the minimum distance at which the variable must be kept with respect to the bound. It can be specified as a value or as **AUTO** in which case, the minimum distance is automatically selected according to the type of variable.
- **PU:** optional physical units

Example:

```
C-1_EFFIC BND_CONTROL Both 0.1 %
```

4.7 DEFINING FILTER SETTINGS

Attention: ValiFilter is available under a separate licence agreement.

ValiFilter detects measured values that are outside **ERROR** bounds (LL-HH interval) and updates their value, precision and failure probability according to user settings. These settings are defined in the model file. They can be updated using one of the two following syntaxes.

```
TAGNAME      FILTER      MEASURED      ACCURACY      FAILPROB
```

where:

- TAGNAME: the Tag name
- MEASURED: the action on the measured value and can be:
 - **NONE**: no action is taken
 - **BND**: the measured value is replaced by the activated bound
 - **DEF**: the measured value is replaced by the default value
 - **MID**: the measured value is replaced by the average of the bounds $(LL+HH)/2$
 - **LAST**: the measured value is replaced by the validated value of the corresponding tag (as obtained during the last successful and saved run)
 - Value Physun: the measured value is replaced by the given value
- ACCURACY: the action on the accuracy and can be:
 - **NONE**: no action is taken
 - **OFF**: the tag is set OFF
 - **DEF**: the accuracy is replaced by the default value
 - **MID**: the accuracy is replaced by half the difference between the bounds $(HH-LL)/2$
 - Value: the accuracy is replaced by the given value (a negative value corresponds to a relative accuracy expressed in %)
- FAILPROB: defines the action on the Failure probability and can be:
 - **NONE**: no action is taken
 - Value %: the failure probability is replaced by the given value

Example:

```
POWER_POWER LL 10 HH 17 MW
POWER_POWER FILTER DEF -5 50 %
AIR.1_T LL -20 HH 40 C
AIR.1_T FILTER BND 10 NONE
```

The second syntax allows to update several tags in one shot by using wildcards:

UPDATETAG	TYPE	TAGTYPE	FILTER	MEASURED	ACCURACY	FAILPROB
UPDATETAG	NAME	TAGNAME	FILTER	MEASURED	ACCURACY	FAILPROB

where:

- TAGTYPE and TAGNAME are strings defining which tags are to be changed (based on their type or name) and can contain wildcards

Example:

```
! Changes
01LCQ01CF001 FILTER 0.1 t/h -5 NONE
! Sets the filter settings of all tags starting by TI to MID MID NONE
!
```

UPDATETAG NAME TI* FILTER MID MID NONE

Attention:

1. The penalty reported in the model and in the result files will be calculated both on basis of the raw measured value as on basis of the filtered value.
2. The control strings and test on CUTOFF value are based on the raw measured value

It is also possible to specify a **CUTOFF** value on the filter settings of an individual tag (ValiFilter license required). If the value of the measurement is below the specified value, the corresponding object (STREAM or UNIT) will be set OFF. The syntax is:

TAGNAME **CUTOFF** VALUE PU
TAGNAME **CUTOFF** **NONE**

where:

- TAGNAME: is the TAG name
- VALUE: is the value under which the stream will be set OFF
- PU: is the physical unit

Attention: It is also possible to specify a CUTOFF value on an individual tag via control strings (see "[Using control strings](#)").

Remember also that the STATUS measurement enables a binary tag value to set a stream OFF.

4.8 DEFINING COMPENSATION SETTINGS

You can update compensations settings of tags through the following syntax

TAGNAME **COMPENS** ID COMPENS BIAS AREF BREF CREF

where:

- TAGNAME: is the tag name
- ID: specifies the compensation mode:
 - 0: none
 - 1: BIAS: bias compensation (using parameters Compens and Bias)
 - 2: DP_TPMW: flow - differential pressure - Ref: T,P,MW
 - 3: DP_TPD0: flow -differential pressure - Ref: T,P,Dens.@0°C
 - 4: VOL_D15: flow -volumetric - Ref: Dens.@15°C,Dens.@T&P,MW
 - 5: DP_D15: flow-differential pressure - Ref: Dens.@15°C,Dens.@T&P,MW
 - 6: DP_D0 : flow-differential pressure - Ref: Dens.@0°C,Dens.@T&P,MW
- COMPENS: is the proportional compensation factor (input in BIAS mode only)
- BIAS: an additive bias
- AREF: first reference parameter (meaning depends on compensation mode)
- BREF: second reference parameter (meaning depends on compensation mode)
- CREF: third reference parameter (meaning depends on compensation mode)

4.9 DEFINING COVARIANCES BETWEEN TAGS

The covariance between two tags allows you to take into account the fact that errors on two different measurements are somehow related, for example because the same sensor is used to measure two different temperatures. In place of defining the covariance, you will define the correlation factors defined as:

$$\text{correl}_{i,j} = \frac{\text{Cov}(x_i, x_j)}{\sigma_i \cdot \sigma_j}$$

The value of this correlation factor must fall between -0.9 and 0.9. On a theoretical basis, the range of values is between -1 and 1. However, on a practical basis values outside -0.9...0.9 are not significant.

The syntax is:

CORREL TAG1 TAG2 VALUE

where:

- TAG1, TAG2: two distinct tag names
- VALUE: covariance value

Example:

```
CORREL TI-001 TI-002 0.1
```

4.10 UNIT PARAMETERS AND INITIAL VALUES OF VARIABLES

Measurements files can also contain values of unit parameters as well as initial values of measured or unmeasured variables.

This allows for example to easily update the MODE parameters of several units without having to click on each of them with ValiModeller GUI.

This feature is also particularly useful when parts of a model have been set OFF and are reset ON. In such cases all streams that were previously OFF have lost their values and must be reinitialised.

Another typical use of this feature is when the plant operates in very different operating conditions. Having different sets of initial conditions might then be necessary to insure a smooth convergence of the model.

The syntax is the following:

```
UNIT  NAME    PARAMETER    VALUE  [PU]
STREAM      NAME    PARAMETER    VALUE  [PU]
```

where:

- NAME: the name of the STREAM or UNIT
- PARAMETER: the name of the parameter
- VALUE: the new value
- PU: optional physical units

The default physical units are the standard BELSIM physical units (K, bar, kmol/s, etc.).

Some useful parameters are:

- UNIT:
 - MODE: calculation mode
 - DP: pressure drop
 - any parameter used by the unit
- MATERIAL STREAM:
 - T: temperature
 - P: pressure
 - VAPF: vapour fraction
 - H: enthalpy
 - xxx: partial molar flow rate of compound xxx
- THERMAL STREAM:
 - LOAD: heat load
- MECHANICAL STREAM:
 - POWER: mechanical power

Attention: The parameter **MODE** must be entered as a numerical value and not as a string. Also for the **REFDP** parameter (reference stream for pressure drop), the stream connection number must be given (positive for an inlet and negative for an outlet stream) and not the stream name.

4.11 PRODUCT ATTRIBUTES

A product attribute must be given to all streams connected to a unit of type NETVAL (see [NETVAL: PIPES NETWORK FOR MASS BALANCE](#) for more information about the representation of networks of pipes through units of type NETVAL).

This attribute (one word of maximum 8 characters) can be set :

- by propagation of the product attribute associated to a tank (TNKVAL) to both its inlet and outlet streams
- by directly specifying the product attribute of the stream in a MEA file
- if no product attribute is specified the 8 first characters of the comment of the stream will be used as an attribute (the comment can be set through ValiModeller GUI or through the COMMENT syntax in a MEA file)

The syntax to set the attribute is:

```
UNIT  NAME  PRODUCT      ATTRIBUTE
STREAM  NAME  PRODUCT      ATTRIBUTE
```

where:

- NAME: the name of the TSREAM or UNIT
- ATTRIBUTE: the product attribute

A specific syntax enables setting OFF all units associated with a given product attribute.

Default properties can also be set on products through the following syntax:

PRODUCT NAME PROPERTY LL HH "COMMENT"

where:

- NAME: the product name
- PROPERTY: the property (e.g. CONVDENS)
- LL, HH: the lower and higher filter bounds of the product property
- COMMENT: a descriptive test of the product, which will be displayed in the product balances in default reports

These default properties of the product can be used to define default values of density measurements set on storage tanks (see section [TNKVAL : STORAGE TANK](#)).

4.12 PRODUCT BALANCES AND IMBALANCES

You can select in which basis the product balances are written in the html report through the syntax:

BALANCE_REPORT TYPE

where:

- TYPE can be MASSF, MOLF, COLDVOLF, CONVVOLF or VOLF

When using product attributes with NETVAL and TNKVAL units, additional tags summarizing the product balances can automatically be created and written to the ValiReport ARC file using the following syntax:

BALANCE **PRODUCT** ATTRIBUTE TYPE

where:

- ATTRIBUTE: is the product attribute
- TYPE: describes the type of tags to be generated and can be:
 - **MASSF**: mass flow rate
 - **MOLF**: molar flow rate
 - **COLDVOLF**: cold volumetric vlow rate (15C)
 - **CONVVOLF**: conventional volumetric flow rate

The following tags are then generated:

- ATTRIBUTE_CHARGE_TYPE: total amount of product sent to storage tanks
- ATTRIBUTE_RUNDOWN_TYPE: total amount of product taken from storage tanks
- ATTRIBUTE_PRODUCT_TYPE: total internal production of product
- ATTRIBUTE_CONSUMP_TYPE: total internal consumption of product
- ATTRIBUTE_IMPORT_TYPE: total import of product
- ATTRIBUTE_EXPORT_TYPE: total export of product
- ATTRIBUTE_OPINVENT_TYPE: opening inventory of product
- ATTRIBUTE_CLINVENT_TYPE: closing inventory of product

If balances are activated by zones (see REPVAL unit), the same kind of tags will be defined for each zone, starting by the zone name. In addition two additional movements type will be defined for the zones: TRAN-IN and TRAN-OUT enabling to trace the transfer between zones.

Attention: the movement names used for these tags can be customized in a **DICTIONARY** section.

Example:

BALANCE PRODUCT C4 MASSF

will automatically generate the following tags:

```
C4_CHARGE_MASSF
C4_RUNDOWN_MASSF
C4_PRODUCT_MASSF
C4_CONSUMP_MASSF
C4_IMPORT_MASSF
C4_EXPORT_MASSF
C4_OPINVENT_MASS
C4_CLINVENT_MASS
```

If zone balancing is activated (and assuming one of the zone is named A1), the following tags will also be generated:

```
A1_C4_CHARGE_MASSF
A1_C4_RUNDOWN_MASSF
A1_C4_PRODUCT_MASSF
A1_C4_CONSUMP_MASSF
A1_C4_IMPORT_MASSF
A1_C4_EXPORT_MASSF
A1_C4_OPINVENT_MASS
A1_C4_CLINVENT_MASS
A1_C4_TRAN-IN_MASSF
A1_C4_TRAN-OUT_MASSF
```

The above is available when no sub-movement is defined anywhere in the application.

When at least one stream has a specific sub-movement, all tag names use the sub-movement convention described below, even for movements without sub-movements.

Assuming the balance is requested in MASSF and the Product is C4, the *.arc file will include some lines like:

```
C4_EXPORT_SHIPMENT_MASSF
C4_EXPORT_LOSSES_MASSF
```

C4_EXPORT_EXPORT_MASSF

When balances by zone are enabled, the following tags will also be generated for each Zone (e.g.A1)::

A1_C4_EXPORT_SHIPMENT_MASSF
A1_C4_EXPORT_LOSSES_MASSF
A1_C4_EXPORT_EXPORT_MASSF

Note that it is possible to change the movement type name through a dictionary (*.mea) file. When replacing EXPORT by EXPORTA through the dictionary, the tags become:

- without sub-movement:

A1_C4_EXPORTA_MASSF

- with sub-movements:

A1_C4_EXPORTA_SHIPMENT_MASSF
A1_C4_EXPORTA_LOSSES_MASSF
A1_C4_EXPORTA_EXPORT_MASSF

The EXPORT in bold here above is the default name of the movement and must be kept as it is.

You can also allow an **imbalance** on a given product by using the following syntax:

IMBALANCE **PRODUCT** ATTRIBUTE TYPE

In such a case, the balance of the selected product will not be closed and an additional tag (ATTRIBUTE_IMBAL_TYPE) is created giving the identified imbalance.

Example:

IMBALANCE PRODUCT C4 MASSF

will automatically generate the following tags:

C4_IMBAL_MASSF

And if zone balancing is activated:

A1_C4_IMBAL_MASSF

4.13 COMMENTS OF OBJECTS

Most objects can have a comment, which can be defined either through ValiModeller GUI or through measurement files.

The syntax is the following:

COMMENT	UNIT	NAME	"COMMENT"
COMMENT	STREAM	NAME	"COMMENT"
COMMENT	THERMOD	NAME	"COMMENT"
COMMENT	COMPOUND	NAME	"COMMENT"
COMMENT	REACTION	NAME	"COMMENT"
COMMENT	TAG	NAME	"COMMENT"

where:

- NAME: the name of the object
- COMMENT: the comment to insert

Attention:

1. If no product attribute is specified on a stream connected to a NETVAL, the 8 first characters of the stream comments are used in place. This feature is kept in the present release by compatibility with older versions.
2. The comments of TAGS (and the control strings they contain if any) can also be updated through the UPDATETAG command

4.14 SETTING UNITS, STREAMS OR PRODUCT "OFF"

Setting a unit or a stream OFF indicates that this element is not operating and that nothing flows through it. The corresponding equations are therefore not generated by VALI. Setting OFF a unit automatically sets OFF all streams connected to that unit. Also any unit (except units of type TNKVAL) having all its inlet material streams or all its outlet material streams OFF is automatically set OFF. Setting OFF a PFD automatically sets OFF all units of the PFD.

Finally setting OFF a PRODUCT (see section "[NETVAL](#)") sets off all units (typically tanks) referring to this product attribute.

Attention: It is also possible to set UNITS OUT, in which case the UNIT is considered by VALI as not existing at all (see below).

The syntax is:

```
UNIT   NAME   OFF
STREAM   NAME   OFF
```

where:

- NAME: the unit (or PFD) or stream name

Another possibility to set OFF units and streams is to use the syntax:

```
PRODUCT   ATTRIBUTE   OFF
```

which will set OFF all tanks associated with the corresponding product Attribute as well as their inlet and outlet streams.

It is also possible to set streams off when a measured value on the streams is below a so-called CUTOFF value. The TAG cannot be related to a compound partial flow rate or fraction but it can be any total rate, temperature or pressure. The CUTOFF values can be set through unit CUTOFF, through the CUTOFF filter setting on individuals tag or through the following syntax is:

```
CUTOFF   VAR   [VALMIN]   VALUE   [PU]
```

where:

- VAR: is the name of the measured variable (e.g. MASSF, MOLF, T, etc.)
- VALMIN: is an optional minimum value under which no action is taken
- VALUE: is the value under which the stream will be set OFF
- PU: is the optional physical unit

Examples:

```
CUTOFF MASSF 0.01 kg/s
! sets OFF any stream for which the measured rate is below 0.01 kg/s
CUTOFF MOLF -1 1 Nm3/h
! sets OFF any stream for which the measured rate is between -1 and 1 Nm3/
```

See "[Defining filter settings](#)" to define CUTOFF on individual tags.

4.15 SETTING UNITS OR PRODUCTS "OUT"

Setting a **unit** OUT indicates that the UNIT is to be considered as not existing. STREAMS connected to this UNIT only are then ignored, while streams connected to other units as well become global inlets or outlets of the model. A complete PFD can also be set OUT. All sub-PFDs and units of this PFD are then ignored by VALI.

This feature is very useful to exclude some parts of a model, which are not yet completed or which are for some reasons creating problems.

Setting a **product** OUT enables to exclude all import and export streams from the validation model and to validate each storage tank independently. This is useful for example when import and export streams measurements are not yet available.

The syntax is:

```
UNIT NAME OUT
PRODUCT NAME OUT
```

where:

- NAME: the name of the unit (or PFD) or product

4.16 THERMOD BINARY PARAMETERS

Measurements files can also contain the values of the binary parameters of the THERMODS.

```
THERMOD NAME PARAMETER COMPOUND1 COMPOUND2 VALUE
```

where:

- NAME: the THERMOD name
- PARAMETER: the binary parameter name
- COMPOUND1: the name of the first compound of the binary
- COMPOUND2: the name of the second compound of the binary
- VALUE: the new value

4.17 COMPOUND DATA

The MEA files can also be used to change the data of Compounds. The syntax is the following:

COMPOUND NAME PARAMETER VALUE

where:

- NAME: is the name of the Compound
- PARAMETER: is the name of the parameter
- VALUE: is the new value

The physical units are the standard BELSIM physical units (K, bar, kmol/s, etc.).

4.18 ELECTRO-NEUTRALITY CONSTRAINTS

The electrical charge of the Compounds (ions) is specified as a Compound property. The following syntax can be used:

COMPOUND NAME **ELECTRO** VALUE

where:

- NAME: is the name of the Compound
- VALUE: is the new value of the electric charge

The electro-neutrality constraint of a STREAM is activated through the syntax:

STREAM NAME **ELECTRO** VALUE

where:

- NAME: the name of the STREAM
- VALUE: the new value of the electric charge

4.19 OUTPUT AND ARCHIVE FILES

PRINCIPLES

Output and Archive files are automatically created by VALI after a successful run and will contain validated values of selected measurements and model variables.

When the run is not successful, the files will by default not be created. If they were already existing before the run they will be deleted.

Attention: It is however possible to force the generation of the file even when the run is not successful by introducing the keyword NOCONVER in any of the measurement files.

To create such a file you must introduce file a first line indicating the type of report (Output or Archive) to be created as well as the name of this report in your measurement. Then, you specify either line by line or as Tables which values must be written to the Output or Archive file.

The only difference between Archive and Output files is the number of values produced for each selected TAG. Output files will include only the validated value while Archive files will include several values for each tag its measured value, measured accuracy, validated value, a posteriori accuracy, physical units, penalty and status.

For other variables (not TAGS), Output and Archive files are the same.

You can generate several Output and/or Archive files in a single measurements file.

Attention:

1. In a given file, all INPUT statements must come before any OUTPUT or ARCHIVE statement.
2. Archive files have been designed to operate with the ValiReport facility.
3. Several OUTPUT/ARCHIVE commands can refer to the same outfile. The different reports will simply be concatenated.
4. The number of different output files may not exceed 16.

COMBINED USE OF INPUT AND OUTPUT OR ARCHIVE FILES

There are some limitations in the combined used of the INPUT and OUTPUT (or ARCHIVE) commands. It is not possible, in a given measurements file, to include files with an INPUT statement after any OUTPUT or ARCHIVE statement.

Examples:

1.

```
INPUT somedata.mea    ! This is a valid combination of INPUT and OUTPUT
INPUT moredata.mea
OUTPUT myreport.res
TI-001
TI-0002
```

2.

```
INPUT somedata.mea    ! This is an invalid combination of INPUT and OUTPUT
OUTPUT myreport.res
TI-001
TI-0002
INPUT moredata.mea
```

In Example 1, it is however perfectly valid to have some OUTPUT sections in the included files (somedata.mea and moredata.mea).

DEFINING THE OUTPUT OR ARCHIVE FILE

The syntax is the same for OUTPUT and ARCHIVE files:

```
OUTPUT [OPTIONS] OUTFILE      OR      ARCHIVE [OPTIONS] OUTFILE
ELEMENT1
ELEMENT2
"STRINGS"
TABLES
...
```

where:

- **OUTFILE**: the name of the Output or Archive File that will be created
Attention: Special characters are not allowed in the name of the Output or Archive File.
- **OPTIONS**: optional formatting commands:
 - -T: fields will be separated by a TAB character
 - -B: fields will be separated by a SPACE character
 - -C: fields will be separated by a comma (,)
 - -S: fields will be separated by a semicolon (;)
 - -A: alphanumeric fields will be enclosed between double quotes ("")
 - -D: uses a comma (,) as decimal symbol
- **ELEMENT1, 2, ETC.**: the various elements to be written in the report (see below): TAG values and UNIT, STREAM, Compound and/or THERMOD data
- **STRINGS**: comments that must be included in the report. The double quotes (") will be removed before writing in the report
- **TABLES**: table specifications (see section "[Writing Tables](#)" below)

MODIFY THE NUMBER OF DECIMALS IN ARCHIVE FILE

By default, ValiEngine prints Tag values in the Archive file with 6 significant digits. In some applications, you may want to retrieve Tag values with more decimals.

This can be defined through a specific text file that will be understood by ValiEngine.

This feature comes with the following limitations:

1. The text file must have the exact name: "DigitsFile.txt",
2. "DigitsFile.txt" must be stored next to the model file (*.bls),
3. You can define the number of digits for the following types of Tags only:
 - MASS, MASSF
 - CONVVOLF, VOL1, VOL2
 - CONVDENS, DENS1, DENS2
4. The maximal number of digits is limited to 9,
5. When writing this file, be advised that the Measurement Type, the Physical Unit and the number of digits columns must be separated by 8 to 16 space characters (see example here below).

Note: tabular characters are not supported.

Example:

```
!-----
!Mass Flowrate
MASSF          t/h          7
MASSF          t/d          6
MASSF          T/H          7
!-----
!Material Mass
MASS           t            6
MASS           TON          6
MASS           T            6
!-----
!Volumetric Flowrate
CONVVOLF       m3/d         6
CONVVOLF       MBD          6
CONVVOLF       m3/h         6
CONVVOLF       bbl/h        6
CONVVOLF       Nm3/h        6
!-----
!Volume
VOL1           m3           6
VOL1           bbl          6
VOL1           BBL          6
VOL2           m3           6
VOL2           bbl          6
VOL2           BBL          6
!-----
!Density
CONVDENS       D60/60       6
CONVDENS       Air_SG       6
CONVDENS       AIR_SG       6
CONVDENS       SG60/60      6
DENS1          SG60/60      6
DENS1          D60/60       6
DENS1          SPGR         6
DENS2          SG60/60      6
DENS2          D60/60       6
DENS2          SPGR         6
```

4.20 WRITING TAG VALUES IN THE REPORT

The values to be written can either be specified one by one (syntax 1), or globally for a number of tags specified using wildcards (syntax 2).

NAME [PU]

DATA TAG [NAME]

where:

- NAME: the name of the TAG to be written
- PU: the physical units for the output (optional). The default physical units defined in the TAG file are used if PU is not given

Attention: TAGS which are off can be specified in the report. A value will be given provided that this TAG can be calculated from the results of the data validation run. A posteriori accuracy will also be given when the TAG is defined on a variable which was actually part of the model.

4.21 WRITING VARIOUS INFORMATION ITEMS IN THE REPORT

A number of keywords can be used to get some values relevant to the run. The syntax is the same for OUTPUT and ARCHIVE files:

KEYWORD ["STRING"]

where:

- **KEYWORD**: one of the following keywords:
 - **PENALTY**: the total sum of penalty
 - **CHISQU**: the chi square value
 - **SSR**: the sum of square of the residues of the constraint equations
 - **QUALITY**: the quality of the run
 - **ITER**: the number of iterations
 - **DATE**: the current date
 - **TIME**: the current time
 - **EXITCODE**: the status of the run
 - **PROJVERS**: the project version as defined in the history
 - **PDB**: the model file name
 - **PROBLEM**: the name of the PFD that has been used for the validation problem
 - **MEAFILE**: the name of the (main) measurement file name
 - **USER**: the user name
 - **VERSION**: the version number of VALI
 - **RUNDATE**: the run data as defined in the input files
 - **INFOI** (i=1..9): some textual information as given in the input files.
 - **NACTBND**: number of active bounds
 - **NREDTOT**: total number of redundancies
 - **NREDTRIV**: number of trivial redundancies
 - **NRED**: number of redundancies
 - **NEQ**: number of equations
 - **NUNMEAS**: number of unmeasured variables
 - **NMEAS**: number of measured variables
 - **NCST**: number of constants
 - **NTAGS**: number of tags (in the model file)
 - **NFLAG**: number of flagged tags
 - **NELIM**: number of eliminated tags
 - **NFILT**: number of filtered tags

- **NINF**: number of INF (info) messages
- **NWRN**: number of WRN (warning) messages
- **NERR**: number of ERR (error) messages
- **NEERS**: number of ERS (severe error) messages
- **NERF**: number of ERF (fatal error) messages
- **NACQISS**: number of acquisition issues
- **NOUTOFRANGE**: number of Tags out of range
- **NRELAX**: number of relaxed Tags
- **NBAD**: number of bad Tags
- **NFLAT1**: number of flat Tags, of level 1
- **NFLAT2**: number of flat Tags, of level 2
- "STRING": a string to be written in place of the keyword name when writing the keyword value.

You can also get the list of units that were part of the validation model by the following syntax:

LIST

4.22 WRITING UNIT, STREAM, COMPOUND AND THERMOD VALUES

The Values to be written can either be specified one by one (syntax 1), or globally for a number of objects of the same type (syntax 2). The first syntax allows to define the contents of your report very precisely. The second syntax is more often used to save the values of all variables so that you can reuse them as starting point. In each case the report produced can generally be reused as an input file.

SYNTAX 1:

UNIT	NAME	PARAMETER	[PU]	["STRING"]
STREAM	NAME	PARAMETER	[PU]	["STRING"]
COMPOUND	NAME	PARAMETER		
THERMOD	NAME	PARAMETER	COMPOUND1	COMPOUND2

where:

- NAME: the name of a UNIT, STREAM, COMPOUND or THERMOD
- PARAMETER: the parameter of the UNIT, STREAM, COMPOUND or THERMOD
- COMPOUND1: the name of the first COMPOUND of the binary
- COMPOUND2: the name of the second COMPOUND of the binary
- PU: the physical units for the output (optional)
- STRING: a string to be written in front of the parameter value (optional)

Some useful parameters are:

- UNIT:
 - MODE: calculation mode
 - DP: pressure drop
 - any parameter of the UNIT
- MATERIAL STREAM:
 - T: temperature
 - P: pressure
 - H: enthalpy
 - xxx: partial flow rate of component xxx
 - VOLF: volumetric flow rate
 - MOLF: molar flow rate
 - MASSF: mass flow rate
 - GMOLF: molar flow rate on a dry basis
 - GMASSF: mass flow rate on a dry basis
 - D15: gravity D15/4
 - D25: gravity D25/4
 - D60: gravity D60/60
 - API: API gravity
 - WFxxx: weight fraction of component xxx
 - MFxxx: molar fraction of component xxx
 - WGxxx: weight fraction of component xxx on dry basis
 - MGxxx: molar fraction of component xxx on dry basis
 - property: derived property (viscosity, specific heat ,etc.)(see section "[CALL PROPTY](#)")
- THERMAL STREAM:
 - LOAD: heat load
- MECHANICAL STREAM:
 - POWER: mechanical power
- COMPOUND:
 - any compound parameter

- THERMOD:
 - any binary parameter

Attention: The report created can be reused as an input file when you do not use any derived stream value (like VOLF and the following) and you do not use "strings".

SYNTAX 2:

```
DATA UNIT [NAME [PARAMETER]]
DATA STREAM [NAME [PARAMETER]]
DATA COMPOUND [NAME [PARAMETER]]
DATA THERMOD [NAME]
```

where:

- NAME: the name of the UNIT, STREAM, COMPOUND or THERMOD. This name can include wildcards (*, ?) in which case all objects matching the string will be selected. If no name is given, all objects are selected
- PARAMETER: the parameter of the UNIT, STREAM, COMPOUND or THERMOD. The parameter name can include wildcards (*, ?) in which case all parameters matching the string will be selected

Valid parameters are:

- UNIT:
 - MODE: calculation mode
 - DP: pressure drop
 - any parameter used by the unit
- MATERIAL STREAM:
 - T: temperature
 - P: pressure
 - VAPF: vapour fraction
 - H: enthalpy
 - xxx: partial molar flow rate of compound xxx
 - property: derived property (viscosity, specific heat ,etc.)(see section "[CALL PROPTY](#)")
- THERMAL STREAM:
 - LOAD: heat load
- MECHANICAL STREAM:
 - POWER: mechanical power
- COMPOUND:
 - any compound parameter
- THERMOD:
 - any binary parameter

4.23 WRITING TABLES

With the TABLE concept, the user can group data of various objects into Tables. To specify the contents of a TABLE several information must be defined:

- the basic layout of the tables. Horizontal tables are tables where one line of information is generated for each selected object. Vertical tables are tables where one column of information is generated for each selected object
- the class of object (UNITS, STREAMS or TAGS) for which the table must be generated
- the order in which objects will be displayed
- criteria for the selection of the objects, typically based on their names or type
- various formatting options

- selection of values (or other type of information) to be produced for each selected object

Although the way the Table information is specified is similar for all classes of objects, some particular features are only available for some classes. The syntax is therefore described per object class.

Some examples including tables are given at the end of this chapter. See also the coal example (directory examples/coal on the distribution CD).

4.23.1 WRITING UNIT TABLES

The syntax is:

```
TABLE {HORIZONTAL|VERTICAL} UNIT
[ORDER_BY {NAME|PFD}]
[SELECTION COMMANDS]
[FORMATTING COMMANDS] OR [VALUE COMMANDS] OR [SUBTABLE COMMANDS]
ENDTABLE
```

Attention: The order command must precede any selection command. All selection commands must appear before any formatting, value or sub-table command

SELECTION COMMANDS

```
SELECT TYPE NAME
SELECT NAME NAME
```

where:

- **name**: either the type of the UNIT or the name of the UNIT. This name can include wildcards (*, ?) in which case all objects matching the string will be selected. If no name is given, all objects are selected

Selection commands can be combined. Selection based on type must appear before any selection based on name. All objects satisfying any of the type selections and any of the name selections will be selected.

FORMATTING COMMANDS

- **MAXCOL I**: maximum number of columns in a vertical table is set to I
- **COLWIDTH I**: minimum width of the columns is set to I. When necessary the width of a column will be automatically increased
- **UNDERLINE "U"**: generates a separation line based on the character U in an horizontal table
- **SEPCHAR "S"**: specifies the character S as the separator between columns
- **NOHEADER**: prohibits the printing of the header line in horizontal tables

VALUE COMMANDS

These commands allow the user to specify which values are to be printed in the table. These parameters can be a parameter of the unit (**PARAM**) or a parameter of an inlet (**INi**) or outlet (**OUTi**) streams, the name of the unit (**NAME**), its comment (**COMMENT**), the name of the PFD where the unit is located (**ORIGPFD**), and its mass balance based on measured values (**MASSBAL**).

PARAM PARAMETER ["STRING"] [PU] [FORMAT]

INi PARAMETER ["STRING"] [PU] [FORMAT]

OUTi PARAMETER ["STRING"] [PU] [FORMAT]

NAME

COMMENT

ORIGPFD

MASSBAL

where:

- **PARAMETER**: parameter of the UNIT or STREAM
- **STRING**: string to be written in place of the parameter name (optional)
- **PU**: physical units for the output (optional)
- **FORMAT**: any valid FORTRAN format specification (ex: (F7.2) prints the value using 7 digits in total, with 2 digits after the decimal point, (G12.4) prints the values in scientific notation using 12 digits in total, with 4 digits after the decimal point) (optional, default format is G12.6)
- **"i" = 1, 2, 3, 4,...** (in **INi** and **OUTi**) : the order of the inlet or outlet streams

Example: In the following example, we will print the temperature parameter of the first 3 inlet and outlet streams of every unit in the model, to a file called "runs-units.txt".

OUTPUT RUNS-UNITS.TXT

TABLE VERTICAL UNIT

NAME

IN1 T

IN2 T

IN3 T

OUT1 T

OUT2 T

OUT3 T

ENDTABLE

4.23.2 WRITING UNIT SUB-TABLES

The concept of sub-table is a way to produce for each unit selected in the unit table section, a secondary report relative to the streams connected to the unit, to its reactions or to the measurements defined on the unit.

A sub-table definition can appear in place of (or in addition to) Value commands in a UNIT Table.

The presence of a sub-table in a Vertical UNIT Table limits to 1 the maximum number of columns of this main Table. The syntax is as follows:

SUBTABLE {**VERTICAL** | **HORIZONTAL**} {**STREAM** | **INLET** | **OUTLET** | **REACTION** | **TAG**}

[**ORDER_BY** {**NAME** | **PFD**}]

[**SELECTION** COMMANDS]

[**FORMATTING COMMANDS**] OR [**VALUE COMMANDS**]
ENDSUBTABLE

SUBTABLE COMMAND

The sub-table commands specifies the table orientation and the objects on which a sub-table will be generated:

- **STREAM**: all connected streams
- **INLET**: all inlet streams
- **OUTLET**: all outlet streams
- **REACTIONS**: reactions
- **TAG**: tags

SELECTION COMMANDS

SELECT TYPE NAME
SELECT NAME NAME

where:

- **NAME**: either the type or the name of the STREAM, REACTION or TAG. This name can include wildcards (*, ?) in which case all objects matching the string will be selected. If no name is given, all objects are selected

Selection commands can be combined. Selections based on type must appear before any selection based on name. All objects satisfying any of the type selections and any of the name selections will be selected.

FORMATTING COMMANDS

- **MAXCOL** I: maximum number of columns in a vertical table is set to I
- **UNDERLINE** "U": generates a separation line based on the character U in an horizontal table
- **SEPCHAR** "S": specifies the character S as the separator between columns
- **NOHEADER**: prohibits the printing of the header line in horizontal tables
- **SUFFIX** "S": specifies the character S as a suffix for all tag names

STREAM VALUE COMMANDS

These commands allow the user to specify which stream values are to be printed in the table. These values can be a parameter of the stream (**PARAM**), its **NAME**, its **COMMENT**, its origin (**FROM**, **PFDFROM**) or destination unit or PFD (**TO**, **PFDTO**), its **STATE** or associated **THERMOD**. The following syntax can be used:

PARAM PARAMETER ["STRING"] [PU] [FORMAT]

```
NAME ["STRING"]  
FROM ["STRING"]  
TO ["STRING"]  
PFDFROM ["STRING"]  
PFDTO ["STRING"]  
STATE ["STRING"]  
THERMOD ["STRING"]  
COMMENT ["STRING"]
```

where:

- **PARAMETER**: parameter of the UNIT or STREAM
- **STRING**: string to be written in place of the parameter name (optional)
- **PU**: physical units for the output (optional)
- **FORMAT**: any valid FORTRAN format specification (ex: (F7.2) prints the value using 7 digits in total, with 2 digits after the decimal point, (G12.4) prints the values in scientific notation using 12 digits in total, with 4 digits after the decimal point) (optional, default format is G12.6)

REACTION VALUE COMMANDS

These commands allow the user to specify which reaction values are to be printed in the table. These values can be a parameter of the stream (**PARAM**), its **NAME**, its **COMMENT**, its origin (**FROM**) or destination unit (**TO**), its **STATE** or associated **THERMOD**.

```
EXTENT ["STRING"] [PU] [FORMAT]  
DTEQ ["STRING"] [PU] [FORMAT]  
NAME ["STRING"]  
COMMENT ["STRING"]
```

where:

- **STRING**: string to be written in place of the parameter name (optional)
- **PU**: physical units for the output (optional)
- **FORMAT**: any valid FORTRAN format specification (ex: (F7.2) prints the value using 7 digits in total, with 2 digits after the decimal point, (G12.4) prints the values in scientific notation using 12 digits in total, with 4 digits after the decimal point) (optional, default format is G12.6)

TAGS VALUE COMMANDS

These commands allow the user to specify which TAGS values are to be printed in the table.

```
KEYWORD ["STRING"] [FORMAT]  
NAME ["STRING"]  
COMMENT ["STRING"]
```

where:

- **KEYWORD**: one of the following keywords:
 - **MEA.VAL**: the measured value
 - **MEA.ACC-**: the measured accuracy (with a negative value when relative)
 - **MEA.ACC%**: the measured accuracy (with a % sign when relative)
 - **REC.VAL**: the validated value
 - **REC.ACC-**: the validated accuracy (with a negative value when relative)
 - **REC.ACC%**: the validated accuracy (with a % sign when relative)
 - **PENALTY**: the penalty
 - **CORRFAC**: the correction factor = $REC.VAL / MEA.VAL$
 - **ATTNFACT**: the attenuation factor = $REC.ACC\% / MEA.ACC\%$
 - **FLAGSTAR**: a * is printed if the measurement is flagged
 - **FLAGNUM**: 0 for unflagged measurement, 1 for flagged measurement
 - **PHYSUN**: the physical units
 - **TYPE**: the type of measurement
 - **OBJTYPE**: the object type (type of unit or stream)
 - **OBJNAME**: the object name
 - **OBJCLASS**: the object class (UNIT / STREAM)
 - **STATUS**: status of the tag (used by ValiReport)
 - **FIL.VAL**: the filtered value
 - **FIL.ACC-**: the filtered accuracy (with a negative value when relative)
 - **FIL.ACC%**: the filtered accuracy (with a % sign when relative)
 - **FIL.PEN**: the filtered penalty
 - **MEA.ACCA**: the measured accuracy (absolute)
 - **REC.ACCA**: the validated accuracy (absolute)
 - **FIL.ACCA**: the filtered accuracy (absolute)
- **STRING**: a string to be written in place of the parameter name (optional)
- **FORMAT**: is any valid FORTRAN format specification (ex: (F7.2) prints the value using 7 digits in total, with 2 digits after the decimal point, (G12.4) prints the values in scientific notation using 12 digits in total, with 4 digits after the decimal point) (optional, default format is G12.6)

4.23.3 WRITING STREAM TABLES

The syntax is:

```
TABLE {HORIZONTAL|VERTICAL} STREAM
[ORDER_BY {NAME|PFD}]
[SELECTION COMMANDS]
[FORMATTING COMMANDS] OR [VALUE COMMANDS]
ENDTABLE
```

Attention: The order command must precede any selection command. All selection commands must appear before any formatting, value or sub-table command

SELECTION COMMANDS

```
SELECT TYPE NAME
SELECT NAME NAME
```

where:

- NAME: either the type (MAT, MECSTRM or THERSTRM) of the STREAM or the name of the STREAM. This name can include wildcards (*, ?) in which case all objects matching the string will be selected. If no name is given, all objects are selected

Selection commands can be combined. Selection based on type must appear before any selection based on name. All objects satisfying any of the Type selections and any of the Name selections will be selected.

FORMATTING COMMANDS

- MAXCOL** I: maximum number of columns in a vertical table is set to I
- COLWIDTH** I: minimum width of the columns is set to I. When necessary the width of a column will be automatically increased
- UNDERLINE** "U": generates a separation line based on the character U in an horizontal table
- SEPCHAR** "S": specifies the character S as the separator between columns
- NOHEADER**: prohibits the printing of the header line in horizontal tables

VALUE COMMANDS

These commands allow the user to specify which stream values are to be printed in the table. These values can be a parameter of the stream (**PARAM**), its **NAME**, its **COMMENT**, its origin (**FROM**, **PFDFROM**) or destination unit or PFD (**TO**, **PFDTO**), its **STATE** or associated **THERMOD**. The following syntax can be used:

```
PARAM PARAMETER ["STRING"] [PU] [FORMAT]
NAME ["STRING"]
FROM ["STRING"]
TO ["STRING"]
PFDFROM ["STRING"]
PFDTO ["STRING"]
```

```

STATE  ["STRING"]
THERMOD ["STRING"]
COMMENT ["STRING"]

```

where:

- **PARAMETER**: parameter of the UNIT or STREAM
- **STRING**: string to be written in place of the parameter name (optional)
- **PU**: physical units for the output (optional)
- **FORMAT**: any valid FORTRAN format specification (ex: (F7.2) prints the value using 7 digits in total, with 2 digits after the decimal point, (G12.4) prints the values in scientific notation using 12 digits in total, with 4 digits after the decimal point) (optional, default format is G12.6)

4.23.4 WRITING TAGS TABLES

The syntax is:

```

TABLE  {HORIZONTAL|VERTICAL} TAG
[UNITOFF]
[STREAMOFF]
[STREAMOUT]
[DUMMY]
[ORDER_BY {NAME|PFD}]
[SELECTION COMMANDS]
[FORMATTING COMMANDS] OR [VALUE COMMANDS]
ENDTABLE

```

Attention: The order command must precede any selection command. All selection commands must appear before any formatting, value or sub-table command.

The **UNITOFF**, **STREAMOFF** and **STREAMOUT** commands allow to include the tags related to streams and units that are OFF or OUT.

The **DUMMY** command allows to include the dummy tags.

SELECTION COMMANDS

```

SELECT TYPE NAME
SELECT UNIT NAME
SELECT STREAM NAME
SELECT NAME NAME

```

where:

- **NAME**: either the type or name of the TAG or the name of UNITS or STREAMS. This name can include wildcards (*, ?) in which case all objects matching the string will be selected. If no name is given, all objects are selected

Selection commands can be combined. Selection based on type must appear before any selection based on unit or stream, which in turn must appear before any selection based on TAG name. All objects satisfying any of the Type, UNIT or STREAM selections and any of the TAG Name selections

will be selected.

FORMATTING COMMANDS

- **MAXCOL** *I*: maximum number of columns in a vertical table is set to *I*
- **COLWIDTH** *I*: minimum width of the columns is set to *I*. When necessary the width of a column will be automatically increased
- **UNDERLINE** "U": generates a separation line based on the character U in an horizontal table
- **SEPCHAR** "S": specifies the character S as the separator between columns
- **NOHEADER**: prohibits the printing of the header line in horizontal tables
- **SUFFIX** "S": specifies the character s as a suffix for all tag names

VALUE COMMANDS

These commands allow the user to specify which TAGS values are to be printed in the table.

KEYWORD ["STRING"] [FORMAT]

NAME ["STRING"]

COMMENT ["STRING"]

where:

- **KEYWORD:** one of the following keywords:
 - **MEA.VAL:** the measured value
 - **MEA.ACC-:** the measured accuracy (with a negative value when relative)
 - **MEA.ACC%:** the measured accuracy (with a % sign when relative)
 - **REC.VAL:** the validated value
 - **REC.ACC-:** the validated accuracy (with a negative value when relative)
 - **REC.ACC%:** the validated accuracy (with a % sign when relative)
 - **PENALTY:** the penalty
 - **CORRFAC:** the correction factor = $REC.VAL / MEA.VAL$
 - **ATTNFACT:** the attenuation factor = $REC.ACC\% / MEA.ACC\%$
 - **FLAGSTAR:** a * is printed if the measurement is flagged
 - **FLAGNUM:** 0 for unflagged measurement, 1 for flagged measurement
 - **PHYSUN:** the physical units
 - **TYPE:** the type of measurement
 - **OBJTYPE:** the object type (type of unit or stream)
 - **OBJNAME:** the object name
 - **OBJCLASS:** the object class (UNIT / STREAM)
 - **STATUS:** status of the tag (used by ValiReport)
 - **FIL.VAL:** the filtered value
 - **FIL.ACC-:** the filtered accuracy (with a negative value when relative)
 - **FIL.ACC%:** the filtered accuracy (with a % sign when relative)
 - **FIL.PEN:** the filtered penalty
 - **MEA.ACCA:** the measured accuracy (absolute)
 - **REC.ACCA:** the validated accuracy (absolute)
 - **FIL.ACCA:** the filtered accuracy (absolute)
- **STRING:** a string to be written in place of the parameter name (optional)
- **FORMAT:** is any valid FORTRAN format specification (ex: (F7.2) prints the value using 7 digits in total, with 2 digits after the decimal point, (G12.4) prints the values in scientific notation using 12 digits in total, with 4 digits after the decimal point) (optional, default format is G12.6)

Attention: When a TAG is set constant (or OFF), the keyword CST (or OFF) will appear in place of CORRFAC and ATTNFACT. If the measured value is 0, the correction factor CORRFAC has no meaning and is replaced by a question mark "?".

4.24 USING IOFILE.EXE TO PROCESS MEASUREMENT FILES

The separate application IOFILE.EXE can be used to process measurements files independently of VALI application. This enables extracting various data from a model or uploading other data into the model without the need to solve any validation problem.

You can start IOFILE.EXE:

- by opening a DOS Command windows, going in the appropriate directory and entering the command:

```
<belsim_dir>\bin\vali.exe [<file>]
```

where <belsim_dir> is the path to your belsim installation directory (by default c:\belsim2001) and the optional <file> keyword is the name of the vif file (pc.vif by default)

- by including this command into a DOS batch file

Attention: The structure and contents of the vif files are described above in the section [Using VALI in Batch Mode](#).

4.25 USING IOFILE.EXE TO CHANGE DEFAULT VALUES

Default values in bls files can be overwritten using IOFILE.EXE, using the option is -DEF. When it is used, the default values of tags are replaced by the values defined in mea files, and status of units or streams are updated according what is defined mea files, and the new bls file is saved after the run.

The following mea file syntax is accepted to change the default values

- Tag_name value accuracy/CST/OFF [Physical_unit]

The default value of the tag Tag_name and its accuracy will be modified.

- Unit_paramater_name value Physical_unit

The value of the parameter Unit_parameter_name will be modified.

- Stream_paramater_name value Physical_unit

The value of the parameter Stream_parameter_name will be modified.

- `UNIT Unit_name ON/OFF/OUT`

The unit `Unit_name` will be set `ON/OFF/OUT`.

- `STREAM Stream_name ON/OFF/OUT`

The stream `Stream_name` will be set `ON/OFF/OUT`.

- `UNIT Unit_name Parameter_name value`

The parameter `Parameter_name` of the unit `Unit_name` will be modified.

- `STREAM Stream_name Parameter_name value`

The parameter `Parameter_name` of the stream `Stream_name` will be modified.

- `COMPOUND Compound_name Parameter_name value`

The parameter `Parameter_name` of the compound `Compound_name` will be modified.

- `UPDATETAG TYPE sometype aproperty value [Physical Unit]`

Or

- `UPDATETAG NAME somename aproperty value [Physical Unit]`

If `aproperty` is `DEF.VAL`, `DEF.ACC`, `RAW.VAL`, `RAW.ACC`, `SL`, `LL`, `L`, `H`, `HH`, `SH`, `FAILPROB`, `COMMENT`, `IS.BAD`, `IS.FLAT1` or, `IS.FLAT2`, the property is changed; if `aproperty` is `MEA.VAL`, `DEF.VAL` is changed; if `aproperty` is `MEA.ACC`, `DEF.ACC` is changed.

Remarks:

- The option `-DEF` with `VALIEXE` is not taken into account;

- If –DEF and –NOS are used in the same time, the BLS file will not be saved;

4.26 EXAMPLE OF MEASUREMENT FILES

Example 1

```

UNIT          HTX2    OFF                ! Set unit HTX2 as not operating
UNIT  HTX1  MODE    1                ! Set unit HTX1 mode parameter to 2
UNIT  HTX1  DP1      0.5 bar          ! Set unit HTX1 first pressure drop to 0.5 bar
UNIT  HTX1  DP2      0.3 bar          ! Set unit HTX1 second pressure drop to 0.3 bar
STREAM S102 T 15 C                    ! Initialise temperature of stream S102 to 15 C
T101   13.2   C                      ! Set Tag T101 measurement value to 13.2 C
P101    3.5   CST                    ! Set Tag P101 measurement value to 3.5 (bar)
                                           ! and consider the variable as a constant

T117   27.5   C                      ! Set Tag T117 measurement value to 27.5 C
P117    2     CST                    ! Set Tag P117 measurement value to 2 (bar)
                                           ! and consider the variable as a constant

INPUT list2.mea                      ! Read measurements file 'list2.mea'
INPUT list3.mea                      ! Read measurements file 'list3.mea'
OUTPUT list3.out                    ! Generate output file 'list3.out'
"Results for March 24th, 1997"      ! Output "Results for March 24th, 1997"
T101  F                      ! Output "T101 ... F"
P101  atm                    ! Output "P101 ... atm"
T117  F                      ! Output "T117 ... F"
P117  atm                    ! Output "P117 ... atm"
STREAM S101 VOLF m3/h "Cooling water:" ! Output "Cooling water: ... m3/h"
UNIT  HTX1  A  m2  "Cooler 1 area:"    ! Output "Cooler 1 area: ... m2"
ARCHIVE list3.arc                  ! Generate output file 'list3.arc'
"Results for March 24th, 1997"      ! Output "Results for March 24th, 1997"
T101  F                      ! Output "T101 ... F"
P101  atm                    ! Output "P101 ... atm"
T117  F                      ! Output "T117 ... F"
P117  atm                    ! Output "P117 ... atm"
STREAM S101 VOLF m3/h "Cooling water:" ! Output "Cooling water: ... m3/h"
UNIT  HTX1  A  m2  "Cooler 1 area:"    ! Output "Cooler 1 area: ... m2"
OUTPUT list4.out                    ! Generate output file 'list4.out'
DATA UNIT * MODE                    ! Writes all MODE parameters
DATA STREAM                        ! Writes all Stream data

```

The resulting output file 'list3.out' could be:

```

Results for March 24th, 1997
T101   54.355 F
P101   3.4542 atm
T117   81.772 F
P117   1.9738 atm
Cooling water:   542.17      m3/h
Cooler 1 area:   119.56      m2

```

The resulting output file 'list3.arc' could be:

```

Results for March 24th, 1997
T101   55.760 3.6000 54.355 1.3654 F
P101   3.4542 0.0000 3.4542 0.0000 atm
T117   81.500 3.6000 81.772 1.0354 F
P117   1.9738 0.0000 1.9738 0.0000 atm
Cooling water:  42.17m3/h
Cooler 1 area:  119.56      m2

```

The resulting output file 'list4.aout' could look like:

```

UNIT  HTX1  MODE    10

```

```

UNIT    HTX2    MODE    10
...
STREAM S101    T      12.419 C
STREAM S101    P      3.4991 bar
STREAM S102    H      ...
...
STREAM S101    T      27.6511      C
STREAM S101    P      2.00000      bar

```

Example 2

The following 'measurement' file creates a file with the definition of all tags relevant to the model. The generated file could be reused as a TAG file. (See in the coal example).

```

OUTPUT tagdef.out
! Beginning of the table of tags
TABLE HORIZONTAL TAG
! Selection commands
UNITOFF
STREAMOFF
SELECT NAME *
! Formatting commands
NOHEADER
COLWIDTH 8
SEPCHAR " "
! Value commands
OBJCLASS
OBJNAME
TYPE
MEA.VAL (G15.6)
MEA.ACC- (G11.3)
PHYSUN
! End of the table
ENDTABLE

```

SECTION 5: THE SENSITIVITY ANALYSIS

This section shows you how to use the tools available for statistical analysis of the results provided by ValiEngine

A default Sensitivity Analysis is performed by setting the VALI Runtime option **-SNS**. The sensitivity analysis is also generated when **-VDI** option is selected. The Sensitivity Analysis tool can however be used on a stand-alone basis to generate more specific results. The files generated by the sensitivity analysis are best exploited through ValiAnalytics GUI.

The sensitivity analysis processes the information available in the validation results to answer the following questions:

- what is the precision of the validated values of the measured and unmeasured variables ? This is called the **validated accuracy (or uncertainty)**. Note that the validated accuracy of measured variables is directly calculated by VALI and shown in the html (or r4v), r5v and r6v files
- what are the measurements that have an effect on the validated uncertainty of variables? These are the **contributors**
- how much a given variable will change if the value of a measurement is changed? This is called the **sensitivity**
- what is the interaction between the measurements: this information is provided by the **covariance** analysis

Having answered these questions allows you to answer a second set of questions:

- what are the measurements that are really improved by the validation? The measurements that have a significant improvement of their accuracy
- what are the measurements that should be improved to improve the accuracy of a given variable? The measurements that have a high contribution to uncertainty of the selected variable
- what is the level of dependency of the measurements ? The measurements that will show a significant covariance factor are with a strong link

This section shows:

- some important preliminary remarks
- the layout of the reports that can be created
- how to start the default Sensitivity Analysis from VALI
- how to use the Sensitivity Analysis independently of VALI
- some theoretical background

5.1 PRELIMINARY REMARKS

The validated accuracy values are based on the accuracy values set on the measurements. Validated accuracy values are not directly affected by the correction applied on some of the measurements. When gross errors are present in the validated values, the validated accuracy estimations are no longer valid.

The analysis should also be considered with caution when bounds are activated. It is probably better to check the pertinence of bounds and take corrective actions: relax bounds or set variables constants to get a model converging without activation of bounds.

The validated accuracy should not be interpreted (nor used) as the validated precision of a measurement sensor. The validated accuracy relates to the validated value of the variable on which the measurement is made and this accuracy takes into account the precision of the sensor itself and the precision of all other measurements defined on other variables that have an effect on this variable. The validated accuracy is in fact the "real" knowledge you have about the variables of the process when all the information (measurements + model) are considered at the same time.

Variables set as constants are not considered in the Sensitivity Analysis. If you want to perform sensitivity analysis on these variables, you must first relax them. Setting a very low uncertainty (but not zero) on the constant will have nearly the same effect as setting it constant.

The bounds activated by the solver (SQPIP) are affecting the results of the sensitivity analysis because they are considered as additional constraints of the system. This means that bounded variables will show a validated accuracy of zero. They will also decrease the uncertainty of related variables.

Attention: Some of the measurements, like rates and compositions, that are specified on STREAMS relate in fact to their Rates record and are handled as such by VALI. You will therefore find measurements on Rates in the sensitivity analysis.

5.2 REPORTS OF THE SENSITIVITY ANALYSIS

Several reports are generated by the Sensitivity Analysis:

- <model>.res: the report you specify
- <model>.sum: a summary report
- <model>.vcv: the report of the covariances between validated values of variables and measurements
- <model>.dcv: the report of the covariances between corrections on measurements and the measurements themselves
- <model>.sds
- <model>_xml.res: xml global report, containing all results. This file is meant to be read by [ValiAnalytics](#)

When running the sensitivity analysis from VALI, using the -SNS runtime option, the report <model>.res will consist of the Variables and Measurements Analysis Reports as described below.

The <model>.vcv will also be generated for the Variables and Measurements. And the <model>.dcv will be generated for all the validated variables.

In case of VDI mode, only the measured variables flagged because of a big difference between measured value and validated value will be considered in the different reports. No <model>.res file will be generated.

The covariance information is useful to identify the cause of gross error because they can be due not to the flagged measurement but to some other which are strongly correlated with those ones.

Afterwards you can also run the Sensitivity Analysis as a stand alone tool, to get some customised reports.

All the examples found in this section are extracted from a modified version of GT example provided with the Release.

VARIABLES ANALYSIS REPORT (<MAIN>.RES)

This report displays for each variable:

- its validated value and accuracy (absolute and relative)
- its measured values and accuracies (absolute and relative) if measurements are defined
- a list of contributors (sorted by decreasing contributions) to its validated inaccuracy
- its sensitivity to the values of his contributors
- the relative gains of accuracy and the penalties of his contributors

Attention: the list of contributors and the corresponding sensitivities can be removed from the report when using the tool in stand-alone mode.

Example of a "Variables analysis" report:

Variable	Tag Name	Value	Abs.Acc.	Rel.Acc.	Penal.	P.U.
POWER	S POWER	Reconciled	16095.	160.09	0.99%	kw
		POWER_POWER	16100.	161.00	1.00%	kw
Measurement	Tag Name	Contrib.	Der.Val.	Rel.Gain	Penal.	P.U.
POWER	S POWER	POWER_POWER	98.87%	0.98867	0.57%	0.00 kw
MFO2	R FUME.1	FUME.2_MFO2	0.75%	-49.519	78.19%	0.04 %
EFFIC	U T-1	T-1_EFFIC	0.23%	1.9002	82.91%	0.02 %
MOLF	R AIR.1	AIR.1_MOLF	0.07%	-0.19757E-03	76.90%	1.81 Nm3/h
MOLF	R NG	NG_MOLF	0.07%	0.36302E-01	4.65%	0.09 Nm3/h
T	S FUME.2	FUME.2_T	0.01%	-0.65920	0.45%	0.00 C

If the variable is not measured, the term validated will be changed in 'Computed' and the list of associated measurements will be empty.

In the case of a variable activating one of its bounds, the term "validated" of the first line will be replaced by 'High Bound' or 'Low Bound'. In this case, the list of variables that are related to the considered variable and have a zero validated accuracy will be listed. The table will report for each identified variable:

- its validated sensitivity to the bound of the variable
- its penalty if the variable was measured

In the case of a variable that has a zero validated accuracy, the contributors identification will be replaced by the identification of the bounds that are influencing the validated value.

MEASUREMENTS ANALYSIS REPORT (<MAIN>.RES)

This report displays for each measurement:

- its validated value and accuracies (absolute and relative)
- its measured value and accuracies (absolute and relative)
- a list of variables the inaccuracy of which is significantly explained by the selected measurement
- the sensitivity of these variables to the value of the selected measurement
- the relative gains of accuracy and the penalties of the variables when they are associated with measurement(s)

Attention: the list of variables and the corresponding sensitivities can be removed from the report when using the tool in stand-alone mode.

Example of a "Measurements analysis" report:

Measurement	Tag Name	Value	Abs.Acc.	Rel.Acc.	Penal. P.U.
POWER	S POWER	Reconciled POWER_POWER	16095. 16100.	160.09 161.00	0.99% 1.00%
					0.00 kw
Variable	Tag Name	Contrib.	Der.Val.	Rel.Gain	Penal. P.U.
POWER	S POWER	POWER_POWER	98.87%	0.98867	0.57%
MFO2	R FUME.1	FUME.2_MFO2	15.59%	-0.14977E-03	78.19%
H	S FUME.1	Computed	14.43%	0.53299	
T	S FUME.1	Computed	14.33%	0.13993E-01	
		FUME.1_T			
EFFIC	U T-1	T-1_EFFIC	7.63%	0.11729E-02	82.91%
MRO2	R FUME.1	Computed	2.34%	-0.78024	
MOLF	R AIR.1	AIR.1_MOLF	1.30%	-3.5234	76.90%
MRN2	R FUME.1	Computed	1.30%	-2.7835	
VAR1	U FLX1	Computed	1.30%	-0.43666E-04	
MRO2	R AIR.1	Computed	1.30%	-0.73990	
MRN2	R AIR.1	Computed	1.30%	-2.7835	

In the case of a variable activating one of its bounds, the term "validated" of the first line will be replaced by 'High Bound' or 'Low Bound'. In this case, the list of variables that are related to the considered variable and have a zero validated accuracy will be listed. The table will report for each identified variable:

- its validated sensitivity to the bound of the variable
- its penalty if the variable was measured

In the case of a variable that has a zero validated accuracy, the contributors identification will be replaced by the identification of the bounds that are influencing the validated value.

COVARIANCE REPORT

The validated accuracies reported for each variable by VALI in its report files and in the reports described in the previous sections are in fact a sub part (the diagonal) of the statistical covariance matrix.

The covariance report is providing the other terms of this matrix.

In all the cases, the measured variables are located at the top of the list and are sorted in decreasing order of penalties. This is done in order to ease the use of the covariance analysis to identify the cause of gross error in a set of measurements. In case of **-VDI** option, only the flagged variables will be reported.

The problem of gross error detection is that the responsible is not necessary one of the flagged variable, but it can be due to another one set with a higher accuracy which is not moving but is forcing the other measured variables of the system to show big deviations. In that case, they will be identified because they will show high correlation level with the flagged variables (Cor.Coeff ~ 1).

It is possible to consider two kind of correlation coefficient analysis. The first one is based on identification of correlation coefficient between the validated values and the measurements values provided to the system. The second kind of analysis is to identify the correlation coefficient between the deviation shown by the measured variables with the measurement values provided to the system. This second kind of analysis is the one suggested by the VDI-2048 recommendation to identify the gross errors.

COVARIANCE ANALYSIS OF VALUES (<MAIN>.VCV)

For each variable value, the covariances with the measurements are provided in tables containing:

- its validated value and accuracy (absolute and relative)
- its measured values and accuracies (absolute and relative) if measurements are defined
- a list of correlated measurements (sorted by decreasing correlation coefficient) with the value of the correlation coefficient and the value of the covariance itself
- the relative gains of accuracy and the penalties of correlated measurement

Example of a "Covariance analysis" report for value of variables:

Measurement		Tag Name	Value	Abs.Acc.	Rel.Acc.	Penal.	P.U.
POWER	S POWER	Reconciled	16095.	160.09	0.99%		kw
		POWER_POWER	16100.	161.00	1.00%	0.00	kw
Measurement		Tag Name	Cor.Coeff.	Covariance	Rel.Gain	Penal.	P.U.
MFO2	R FUME.1	FUME.2_MFO2	-39.71%	-0.38823E-01	78.19%	0.04	%
EFFIC	U T-1	T-1_EFFIC	27.78%	0.30403	82.91%	0.02	%
MOLF	R AIR.1	AIR.1_MOLF	-11.49%	-1.1319	76.90%	1.81	Nm3/h
MOLF	R NG	NG_MOLF	2.85%	0.64785E-02	4.65%	0.09	Nm3/h
T	S FUME.2	FUME.2_T	-0.83%	-2.6368	0.45%	0.00	C
T	S AIR.1	AIR.1_T	0.46%	0.73380	0.59%	0.00	C
EFFIC	U C-1	C-1_EFFIC	0.46%	0.82710E-02	71.12%	0.11	%
P	S FUME.2	FUME.2_P	-0.44%	-0.14195E-01	0.41%	0.00	bar
P	S AIR.2	AIR.2_P	0.42%	0.26207	2.08%	0.01	bar
T	S AIR.2	AIR.2_T	-0.15%	-0.49022	0.38%	0.00	C

If the variable is not measured, the term validated will be changed in 'Computed' and the list of associated measurements will be empty.

In the case of a variable activating one of its bounds, the term "validated" of the first line will be replaced by 'High Bound' or 'Low Bound'.

In the case of a variable that has a zero validated accuracy, the table of related measurements will be removed because a trivial value of zero is associated with all the measurements.

The trivial correlation factor of 1 of the variable with itself has been removed from the table.

COVARIANCE ANALYSIS OF DEVIATIONS (<MAIN>.DCV)

For each measurement correction, the covariances with the measurements are provided in tables containing:

- its validated value and accuracy (absolute and relative)
- its measured value and accuracy (absolute and relative) if measurements are defined
- a list of correlated measurements (sorted by decreasing correlation coefficient) with the value of the correlation coefficient and the value of the covariance itself
- the relative gains of accuracy and the penalties of correlated measurements

Attention: Only measured variables are possible in this report file.

Example of a Covariance Analysis report for correction of measurements:

Measurement		Tag Name	Value	Abs.Acc.	Rel.Acc.	Penal.	P.U.
POWER	S POWER	Reconciled	16095.	160.09	0.99%		kw
		POWER_POWER	16100.	161.00	1.00%	0.00	kw
Correction		Tag Name	Cor.Coeff.	Covariance	Rel.Gain	Penal.	P.U.
T	S NG	NG_T	-93.73%	-0.24952E-01	0.00%	0.00	C
MOLF	R NG	NG_MOLF	-84.34%	-0.64785E-02	4.65%	0.09	Nm3/h
MFO2	R FUME.1	FUME.2_MFO2	82.91%	0.38823E-01	78.19%	0.04	%
T	S FUME.2	FUME.2_T	80.76%	2.6368	0.45%	0.00	C
DP	U R-1	R-1_DP	45.02%	0.17409E-03	0.00%	0.00	bar
EFFIC	U T-1	T-1_EFFIC	-45.02%	-0.30403	82.91%	0.02	%
P	S FUME.2	FUME.2_P	45.01%	0.14195E-01	0.41%	0.00	bar
T	S AIR.1	AIR.1_T	-39.52%	-0.73380	0.59%	0.00	C
MOLF	R AIR.1	AIR.1_MOLF	25.48%	1.1319	76.90%	1.81	Nm3/h
P	S AIR.2	AIR.2_P	-18.86%	-0.26207	2.08%	0.01	bar

SUMMARY REPORTS (<MAIN>.SUM)

The default report displays for each variable, starting with the measured ones:

- its validated value and accuracies (absolute and relative)
- the number of measurements (Mea.)
- its measured value and accuracies (absolute and relative), plus relative gain of accuracy and penalty if measurements are defined

Example of a "Summary" report:

Variable		Mea.	Value	Abs.Acc.	Rel.Acc.	Rel.Gain	Penal.	P.U.
S NG	P		20.000	0.40000	2.00%			bar
		1	20.000	0.40000	2.00%	0.00%	0.00	
S NG	T		15.000	1.0000	0.35%			C
		1	15.000	1.0000	0.35%	0.00%	0.00	
R FUME.1	MFO2		14.058	0.61076E-01	0.43%			%
		1	14.000	0.28000	2.00%	78.19%	0.04	
S FUME.2	T		599.95	1.9909	0.23%			C
		1	600.00	2.0000	0.23%	0.45%	0.00	
S AIR.2	T		499.93	1.9923	0.26%			C
		1	500.00	2.0000	0.26%	0.38%	0.00	
R AIR.1	MOLF		0.18610E+06	4967.2	2.67%			Nm3/h
		1	0.21500E+06	21500.	10.00%	76.90%	1.81	
U T-1	EFFIC		80.524	0.68375	0.85%			%
		1	80.000	4.0000	5.00%	82.91%	0.02	

In case of a variable fixed by one of its bound, 'H' or 'L' will appear in the column for measurements count (first line of variable) to show the active bound.

SPREAD-SHEET COMPATIBLE SUMMARY REPORT

This displays for each variable, starting with the measured ones:

- its validated value and accuracies (absolute and relative)
- the number of measurements (Mea.)
- its measured value and accuracies (absolute and relative), plus relative gain of accuracy and penalty if measurements are defined

Example of a "Spread-sheet compatible" report:

Type	OBJECT	Parameter	NBR.MEA.	MEA.VAL.	REC.VAL.	MEA.ACC.	REL.	REC.ACC.	REL.	REL.GAIN	PENAL.	P.U.
R	GN	MOLF	1	8000.0	8151.8	300.00	0.500E-01	170.70	0.277E-01	0.43099	0.25490	Nm3/h
S	FUME.2	T	1	1029.0	1029.2	5.0000	0.385E-02	4.4474	0.343E-02	0.11051	0.219E-02	C

R	FUME.1	OR			0.11238			0.396E-02	0.382E-01			kmol/s
S	PORT	POWER			30110.			798.51	0.265E-01			kW

The main difference with the normal report is that all the information is now on one single line.

In case of variable fixed by one of its bound, flag 'H' or 'L' will be raised in the column following the physical units.

5.3 HOW TO PERFORM A SENSITIVITY ANALYSIS

To generate the Sensitivity Analysis reports, you must first run your Validation model with the option **-SNS**. The intermediate file required for the sensitivity analysis will also be generated when **-VDI** option is used.

VALI then generates <model>.fmt where <model> is the name of the main PFD of your model. This file contains the information required for the Sensitivity Analysis

. It also generates several default reports:

- <model>.res: variables and measurements analysis (not generated with -VDI option)
- <model>.vcv: variables values covariance analysis
- <model>.dcv: measurements corrections covariance analysis
- <model>.sum: summary report
- <model>.sds
- <modell>_xml.res: xml global report, containing all results. This file is meant to be read by [ValiAnalytics](#)

If you want to customize the report contained in <model>.res, you must then use the programme called Sensiv.exe located in the \bin directory of Belsim installation. The following steps must be completed:

- specify the model you want to analyse
- define the settings of the analysis:
 - variables to be analysed
 - measurements to be analysed
 - options
- run the Sensitivity Analysis
- browse your report

STEP 1 : SPECIFYING THE MODEL TO BE ANALYZED

The model to be analysed is either specified in the file sensitiv.vif or directly when running the sensitivity analysis.

The file sensitiv.vif has 4 lines:

- line 1: S to define the analysis from the screen (step 2), F to read the settings of the analysis from the file <model>.com
- line 2: name of the model to be analysed
- line 3: code for the spreadsheet settings: code = SEP + DEC + ALP:
 - SEP: the ASCII code of the field separator (from 0 to 127)
 - DEC:
 - 1000 if the decimal character is the comma
 - 0 if the decimal character is the dot
 - ALP:
 - 2000 if text fields must be enclosed between quotes

- 0 otherwise
- line 4: line for options, actually, the only option recognized is the -VDI which can be used in order to generate a set of report similar as when -VDI option is used to run the model in VALI. The default is to keep a blank line to get the default reports covering all the variables.

STEP 2 : DEFINING THE SENSITIVITY ANALYSIS

The settings of the analysis can be specified either in the file **<model>.com** or directly when running the sensitivity analysis . (<model> stands for the main PFD of the model.)

Depending on the selections you have made, three different reports are produced:

- selecting neither variables nor measurements produces the **default report** with measured and validated values and accuracies of all variables
- selecting only variables generates the **variables analysis report** with measured and validated values and accuracies of the selected variables. On request, measurements contributing to the validated variances are listed with optionally the sensitivity of selected variables to the measurements value
- selecting only measurements produces the **measurements analysis report** with the list of variables that have a significant part of their validated variance due to the accuracy of the selected measurements
- selecting both variables and measurements allows you to get a **variable analysis report** showing the sensitivities of variables to measurements values even when their contribution are low (or zero)

In all the cases, the summary report file (<model.sum>) and the covariance report files (<model>.vcv and <model>.dcv) will be filled at the end of the process taking into account the relevant options used for the definition of the last section of the command file.

In the case of existing command file in the directory used to run VALI, the file is not replaced allowing the use of personal settings to overwrite defaults.

The file **<model>.com** can contain the following commands:

- **:INCLUDE <FILENAME>:** includes commands from the file <filename>
- **:OUTPUT <FILENAME>:** print the analysis results of the current section in the file <filename>. By default, the report file is <model>.res
- **:SPREAD:** when following the definition of an output file leads to generation of all the following reports if spreadsheet format

Trick : modify the default **<model>.com** file introducing the line **:SPREAD** after the line **:OUTPUT <model>.res** will lead to reports in spreadsheet format.

- **VARIABLE** TYP NAME VAR: defines the variables to be analysed (one line per variable):
 - TYP: the type of object (UNIT, STREAM)
 - NAME: the name of the object
 - VAR: the name of the parameter of the object (T, P, MASSF, etc.)

Wildcards (*, ?) can be used in any of the fields.

Example:

```
VARIABLE      STREAM IN      MASSF
```

- **MEASUREMENT** TAGNAME: defines the measurements to be analysed (one line per measurement)
- TAGNAME: the name of the TAG defining the measurement. Wildcards (*, ?) can be used

Example:

```
MEASUREMENT    IN_MASSF
```

- **NUMCONT** N
 - N: the maximum number of contributors to be shown for each measurement or variable. This parameter will also be considered for the covariance analysis
- **MINCONT** VAL: contributors with a value lower than VAL are excluded from the list. This parameter will also be considered for the covariance analysis
- **SENSCONT** Y: forces the display of the sensitivities for each contributor
- **END**: identifies the end of a section and initiates the analysis

STEP 3: RUNNING THE SENSITIVITY ANALYSIS

To run the sensitivity analysis, you must run the program Sensiv.exe located in the \bin directory of your Belsim installation. The easiest way to use it is to define a short-cut to it on your desktop and to specify your work directory as the Start-up directory.

If you have not defined the model in step1, Sensiv will prompt you for the relevant information.

If you have not defined the analysis in step2, Sensiv will also prompt you for the relevant information. You can at this stage ask to create the file defining the analysis so that you can reuse it later on.

STEP 4: BROWSING THE REPORT FILES

The results files can be browsed directly with any text editor. The spread-sheet formatted files can also be imported in any commercial spread-sheet package.

5.4 THEORETICAL BACKGROUND

The following theory is developed for a single measurement on each measured variable. It can be extended to multiple measurements cases by first computing weighted average values for measured data (both measurement's value and measurement's accuracy).

In case of non diagonal covariance matrix (a priori covariance coefficients provided by user), the multiple measurements are avoided by creating additional variable and link equation.

NOTATIONS

$\underline{Y}$	array of the MES corrected or validated measurements.
$\underline{\bar{y}}$	array of the MES average measured values.
$\underline{\sigma^2}$	array of the MES average variance associated with each measurement.
$\underline{X}$	array of the NMES unmeasured variables to be calculated.
$\underline{F(X,Y)}$	array of NEQ constraint equations (balance, equilibrium and link equations).

SOLVING THE VALIDATION PROBLEM

The constraint equations will iteratively be linearised around $\underline{k}^{\text{th}}$ successive approximation of $(\underline{X}^k, \underline{Y}^k)$ and can thus be expressed as:

$$\underline{\underline{A}} \underline{Y} + \underline{\underline{B}} \underline{X} + \underline{\underline{C}} = 0$$

where:

- $\underline{\underline{A}}$ is a (NEQ x MES) constant coefficient matrix
- $\underline{\underline{B}}$ is a (NEQ x NMES) constant coefficient matrix
- $\underline{\underline{C}}$ is a (NEQ) constant coefficient vector

The validation problem can be expressed as a constrained minimisation problem:

$$\min_{\underline{X}, \underline{Y}} (\underline{Y} - \underline{\bar{y}})^T \underline{\underline{P}} (\underline{Y} - \underline{\bar{y}})$$

$$\text{s.t. } \underline{F(X,Y)} = 0$$

with $\underline{\underline{P}}$ the weighting matrix which enables the relative precision of the MES measurements to be quantified.

As a matter of fact, this matrix is the inverse of the a priori covariance matrix

$$\underline{\underline{P}} = \underline{\underline{S_y}}^{-1}$$

$$= \begin{bmatrix} \sigma_1^2 & \rho_{21}\sigma_1\sigma_2 & \dots & \rho_{MES1}\sigma_1\sigma_{MES} \\ \rho_{12}\sigma_1\sigma_2 & \sigma_2^2 & \dots & \rho_{MES2}\sigma_2\sigma_{MES} \\ \dots & \dots & \dots & \dots \\ \rho_{1MES}\sigma_1\sigma_{MES} & \rho_{2MES}\sigma_2\sigma_{MES} & \dots & \sigma_{MES}^2 \end{bmatrix}^{-1}$$

If the covariance elements are equal to 0, the matrix $\underline{\underline{P}}$ is diagonal and its elements are the inverse of variance σ_i^2 of the measurements:

$$\underline{\underline{P}} = \text{diag}\left(\frac{1}{\sigma_i^2}\right)$$

The definition of the covariance matrix is made according to the VDI 2048 specification.

The constrained problem can be transformed into an unconstrained one by using Lagrange formulation (equivalent to the formula 114 of VDI specifications):

$$\min_{\underline{\underline{X}}, \underline{\underline{Y}}, \underline{\underline{\lambda}}} L = (\underline{\underline{Y}} - \underline{\underline{y}})^T \underline{\underline{P}} (\underline{\underline{Y}} - \underline{\underline{y}}) + 2 \underline{\underline{\lambda}}^T \underline{\underline{F}}(\underline{\underline{X}}, \underline{\underline{Y}})$$

where $\underline{\underline{\lambda}}$ is a (NEQ) vector (Lagrange multipliers).

Stationary conditions are:

$$\begin{aligned} \underline{\underline{P}} \underline{\underline{Y}} &+ \underline{\underline{A}}^T \underline{\underline{\lambda}} &= \underline{\underline{P}} \underline{\underline{y}} \\ &+ \underline{\underline{B}}^T \underline{\underline{\lambda}} &= 0 \\ \underline{\underline{A}} \underline{\underline{Y}} &+ \underline{\underline{B}} \underline{\underline{X}} &= -\underline{\underline{C}} \end{aligned}$$

These conditions can be formally expressed as:

$$\underline{\underline{E}} = \underline{\underline{J}}^{-1} \underline{\underline{D}}$$

by defining the total Jacobian matrix $\underline{\underline{J}}$ and the array $\underline{\underline{E}}$ and $\underline{\underline{D}}$:

$$\underline{\underline{J}} = \begin{bmatrix} \underline{\underline{P}} & \underline{\underline{0}} & \underline{\underline{A}}^T \\ \underline{\underline{0}} & \underline{\underline{0}} & \underline{\underline{B}}^T \\ \underline{\underline{A}} & \underline{\underline{B}} & \underline{\underline{0}} \end{bmatrix} \quad \underline{\underline{E}} = \begin{bmatrix} \underline{\underline{Y}} \\ \underline{\underline{X}} \\ \underline{\underline{\lambda}} \end{bmatrix} \quad \underline{\underline{D}} = \begin{bmatrix} \underline{\underline{P}} \underline{\underline{y}} \\ \underline{\underline{0}} \\ -\underline{\underline{C}} \end{bmatrix}$$

Both $\underline{Y}$ and $\underline{X}$ arrays appear as linear combinations of array of measurement data $\overline{\underline{y}}$.

$$\begin{aligned}
 Y_i &= \sum_{l=1}^{MES+NMES+NEQ} J^{-1}_{il} D_l \\
 &= \sum_{j=1}^{MES} \left(J^{-1}_{ij} \sum_{k=1}^{MES} P_{jk} \overline{y_k} \right) - \sum_{k=1}^{NEQ} J^{-1}_{iMES+NMES+k} C_k \\
 &= \sum_{k=1}^{MES} \left(\sum_{j=1}^{MES} J^{-1}_{ij} P_{jk} \right) \overline{y_k} - \sum_{k=1}^{NEQ} J^{-1}_{iMES+NMES+k} C_k
 \end{aligned}
 \quad \forall i=1, \dots, MES$$

In these expressions, matrices A and B (hidden in J-1) are calculated after convergence (solution) of the validation problem.

THE COVARIANCE MATRIX OF VALIDATED MEASUREMENTS

Considering a function (or array of function) defined from the linear combination of the different variables measured and known with a defined uncertainty, it is possible to estimate the variance of the result values :

$$\underline{\underline{g}} = \left(\frac{\partial \underline{g}}{\partial \underline{s}} \right) \times \underline{\underline{s}} \quad \text{and} \quad \underline{\underline{S_z}} = \left(\frac{\partial \underline{g}}{\partial \underline{s}} \right) \times \underline{\underline{S_s}} \times \left(\frac{\partial \underline{g}}{\partial \underline{s}} \right)^T$$

where:

- $\frac{\partial \underline{g}}{\partial \underline{s}}$ is the matrix of linear coefficient, which can also be called sensitivity matrix
- $\underline{\underline{S_s}}$ is the covariance matrix of the measurements
- $\underline{\underline{S_g}}$ is the covariance of the derived functions

This relation is similar to the equation 192 of the VDI 2048 specifications which has been used for computation of the guarantee conditions.

Now, if we consider the functions linking the validated values of measured variables with the values of measurements (see above), the covariance matrix of validated values can be expressed as:

$$\underline{\underline{S_Y}} = \underline{\underline{J^{-1}_{1,MES}}} \times \underline{\underline{P}} \times \underline{\underline{S_{\underline{y}}}} \times \left(\underline{\underline{J^{-1}_{1,MES}}} \times \underline{\underline{P}} \right)^T = \underline{\underline{J^{-1}_{1,MES}}} \times \underline{\underline{P}} \times \underline{\underline{J^{-1}_{1,MES}}} = \underline{\underline{J^{-1}_{1,MES}}} \times \underline{\underline{S_{\underline{y}}^{-1}}} \times \underline{\underline{J^{-1}_{1,MES}}}$$

where:

- $J^{-1}_{1,MES}$ is built from the 1 to MES first lines and contains MES columns

The formula is also taking into accounts the fact that both J-1 and P are symmetrical matrices, and that P is the inverse of the a priori covariance matrix.

With this procedure, it is possible to define a posteriori accuracy for measured and variables.

Actually, VALI is computing only a posteriori variances (s^2) which are converted to standard deviation (s) and reported to the user. This means that VALI is computing only diagonal elements of the a posteriori covariance matrix.

Extension to off diagonal elements has been made in the external programme SENSIV and is reported in a separate report file available only when sensitivity analysis is made. In fact, we are reporting the covariance factor defined from the following formula:

$$\rho_{Y_{ij}} = \frac{S_{Y_{ij}}}{\sqrt{S_{Y_{ii}}S_{Y_{jj}}}}$$

where:

- q is the matrix of the correlation coefficients

The values are in the range -1, +1. Absolute value close to 1 means a strong interaction between the variables. This can help for detection of gross error by showing which measurements are conflicting.

It is possible to identify the covariance matrix of the deviation of the measured variables considering the following definition for the deviations: $\underline{v} = \underline{Y} - \underline{\bar{y}}$

In VALI default mode, the a posteriori values reported for the correlation coefficients are relative to the validated values of the measurements. When the VDI mode is selected, the a posteriori covariances reported are relative to the deviations between the validated value and the value of the measurement.

The covariance matrix of the deviations can be calculated from the following relationships coming from formulas 130 & 132 of VDI 2048 specifications:

$$\underline{v} = \underline{Y} - \underline{\bar{y}} \quad \text{and} \quad \underline{\underline{S_v}} = \underline{\underline{S_y}} - \underline{\underline{S_Y}}$$

where:

- $\underline{v}$ is the improvement vector

The test suggested by formula 129 of the VDI 2048 specifications is based on the same definition of the correlation coefficient which leads to:

$$\rho_{v_{ij}} = \frac{S_{v_{ij}}}{\sqrt{S_{v_{ii}}S_{v_{jj}}}}$$

Another main difference between the VALI and the VDI mode is that only measured variables suspected to be linked with gross error are reported. In the latter case the criterion used for the detection of measured variables to be reported is coming from the formula 141 of the VDI 2048 specifications:

$$\left| \frac{v_i}{\sqrt{\max\left(S_{v,i}, \frac{S_{y,i}}{10}\right)}} \right| > 1.96$$

The value of the chi-square with a statistical certainty of 95 % is reported (in case of VALI mode, the certainty level can be changed). When the value of the objective function is greater than this value, a flag is raised.

The criteria is (VDI 2048 specifications formula 126):

$$(\underline{Y} - \underline{\bar{y}})^T \underline{P}(\underline{Y} - \underline{\bar{y}}) \leq \chi^2_{r,95\%}$$

where :

- r is the redundancy of the problem defined as the number of measurements coming in addition to the number strictly necessary to get a calculable system. When bounds are activated, the number of active bounds is added to the redundancy

It is then possible to identify the suspected measurements from reports and to generate a covariance analysis to identify the correlated variables (VDI 2048 formulas 141 and 129).

HANDLING OF BOUNDS

When using SQPIP solver, it is possible to handle bounds on a rigorous way. The solver is an optimiser designed to handle bounds for all the variables of the system. The problem solved is now:

$$\min_{\underline{X}, \underline{Y}} (\underline{Y} - \underline{\bar{y}})^T \underline{P}(\underline{Y} - \underline{\bar{y}})$$

$$\text{s.t } \underline{F}(\underline{X}, \underline{Y}) = 0$$

$$\underline{Y}_{\min} \leq \underline{Y} \leq \underline{Y}_{\max}$$

$$\underline{X}_{\min} \leq \underline{X} \leq \underline{X}_{\max}$$

For the calculation of the a posteriori covariance matrix, we consider that the variables that are activating the Interior Point procedure of the solver can be considered as constant fixed to the bound value.

This means that for the a posteriori analysis, we are considering that the following equivalent problem is solved:

$$\begin{aligned} \min_{\underline{X}, \underline{Y}} (\underline{Y} - \underline{\bar{y}})^T \underline{P} (\underline{Y} - \underline{\bar{y}}) \\ \text{s.t. } \underline{F}(\underline{X}, \underline{Y}) = 0 \\ \underline{Y}_i = \underline{Y}_{iact} \\ \underline{X}_i = \underline{X}_{iact} \end{aligned}$$

The additional set of constraints expresses that the corresponding variables can be considered as constant and are equal to the activated bound (upper or lower). Practically, this means that we add a new line to the $\underline{A}$ matrix (for a bound on measured variable) or $\underline{P}$ matrix (for a bound on unmeasured variable).

THE COVARIANCE MATRIX OF UNMEASURED VARIABLES

The calculation of a posteriori covariance matrix can be extended to unmeasured variable considering the linearised relationship with the measurement.

$$\begin{aligned} \underline{X}_i &= \sum_{l=1}^{MES+NMES+NEQ} J^{-1}_{MES+1l} D_l \\ &= \sum_{j=1}^{MES} \left(J^{-1}_{MES+1j} \sum_{k=1}^{MES} P_{jk} \bar{y}_k \right) - \sum_{k=1}^{NEQ} J^{-1}_{MES+1MES+NMES+k} C_k \\ &= \sum_{k=1}^{MES} \left(\sum_{j=1}^{MES} J^{-1}_{MES+1j} P_{jk} \right) \bar{y}_k - \sum_{k=1}^{NEQ} J^{-1}_{MES+1MES+NMES+k} C_k \end{aligned} \quad \forall i=1, NMES$$

Considering the equation 192 of the VDI 2048 specifications for the previous functions linking the unmeasured variables with the values of measurements, the covariance of the unmeasured variables can be expressed as:

$$\underline{S}_X = \underline{J}^{-1}_{MES+1MES+NMES} \times \underline{P} \times \underline{S}_Y \times \left(\underline{J}^{-1}_{MES+1MES+NMES} \times \underline{P} \right)^T = \underline{J}^{-1}_{MES+1MES+NMES} \times \underline{P} \times \underline{J}^{-1}_{MES+1MES+NMES}^T$$

where:

- $\underline{J}^{-1}_{MES+1MES+NMES}$ is built from the MES+1 to MES+NMES first lines and contains MES columns

The formula is also taking into accounts the fact that P is symmetrical matrix and is inverse of a priori covariance matrix.

VALI is reporting the a posteriori variances (σ^2) which are converted to standard deviations (σ). The off diagonal elements are representing the a posteriori covariance matrix between the unmeasured variables.

The overall a posteriori covariance matrix can be expressed by the following formula:

$$\underline{\underline{S}} = \underline{\underline{J}}^{-1} \times \underline{\underline{P}}' \times \underline{\underline{J}}^{-1} = \begin{bmatrix} \underline{\underline{S}}_Y & \underline{\underline{S}}_{YX} \\ \underline{\underline{S}}_{XY} & \underline{\underline{S}}_X \end{bmatrix} = \begin{bmatrix} \underline{\underline{S}}_Y & \underline{\underline{S}}_{XY}^T \\ \underline{\underline{S}}_{XY} & \underline{\underline{S}}_X \end{bmatrix}$$

where:

$$\underline{\underline{P}}' = \begin{bmatrix} \underline{\underline{P}} & \underline{\underline{0}} \\ \underline{\underline{0}} & \underline{\underline{0}} \end{bmatrix} \text{ is a square matrix of dimension MES + NMES.}$$

This formula is obtained considering that unmeasured variables have a standard deviation equal to infinite (This leads to the zero values around the $\underline{\underline{P}}$ matrix). We can define a priori covariance

matrix equal to $\underline{\underline{S}}'_X = \begin{bmatrix} \underline{\underline{S}}_Y & \underline{\underline{0}} \\ \underline{\underline{0}} & \infty \times \underline{\underline{I}} \end{bmatrix}$ and we can also consider the limit following behaviour:
 $\underline{\underline{P}}' \times \underline{\underline{S}}'_X = \underline{\underline{I}}$

It is therefore possible to express the covariance matrix of the unmeasured variables with the measured ones according to the following formula:

$$\underline{\underline{S}}_{XY} = \underline{\underline{J}}^{-1}_{MES+1,MES+NMES} \times \underline{\underline{P}} \times \underline{\underline{J}}^{-1}_{1,MES+1} = \underline{\underline{J}}^{-1}_{MES+1,MES+NMES} \times \underline{\underline{S}}_Y^{-1} \times \underline{\underline{J}}^{-1}_{1,MES+1}$$

This formula is used in the external application, which is performing the extended sensitivity analysis (SENSIV). Once again, the programme is reporting the correlation coefficients.

MULTIPLE MEASUREMENTS ON A SINGLE VARIABLE

The theory described here above can be extended to multiple measurements cases by first computing weighted average values for measured data (both measurement's value and measurement's standard deviation).

In case of non diagonal covariance matrix (a priori covariance coefficients provided by user), the multiple measurements are avoided by creating additional variable and link equation.

Weighted average values differ from the usual arithmetic average. The formula below gives more weight to the most accurate measurements.

$$\bar{y}_i = \frac{\sum_{j=1}^{nti} \left(\frac{y_{ij}}{\sigma^2_{ij}} \right)}{\sum_{j=1}^{nti} \left(\frac{1}{\sigma^2_{ij}} \right)} \quad \forall i=1, \text{MES}$$

$$\sigma_i^2 = \frac{1}{\sum_{j=1}^{nti} \left(\frac{1}{\sigma^2_{ij}} \right)}$$

with n_{ti} the number of measurements associated with the measured variable i and j the index of this measurement in the subset of measurements corresponding to the variable.

SPECIFICITY OF VDI MODE

When VALI is running in VDI mode, the precision given to each measurement is expressed in terms of 95% confidence interval, rather than in terms of standard deviations. The 95% confidence interval size is used for both input and outputs by the programme. The relationship between the two values is a simple conversion factor:

$$\bar{V} = 1.96 \times \bar{\sigma}$$

The second specific point is that all the covariance analysis is made for the deviations of the measured variables (v). The analysis with the VALI mode is made for the covariances between the validated values.

The two cases are identified by the titles of the tables presented in the report.

With the VDI mode, the first table is named 'Measurement' and the second is named 'Corrections'. In VALI mode, the title of the first table is 'Variable' and the title of the second one is 'Measurement'.

The last difference between values reported in the tables is the definition of the penalties. In case of VDI mode, the penalty is defined from the following definition that has been provided by the VDI 2048 specifications (formula 141 is using the square root of the penalty reported):

$$\left(\frac{v_i}{\sqrt{\max\left(S_{vi,i}, \frac{S_{\bar{v},i}}{10}\right)}} \right)^2$$

In VALI mode, the penalty reported is corresponding to the following definition:

$$\left(\frac{v_i}{\bar{\sigma}_i} \right)^2 = \left(\frac{v_i}{\sqrt{S_{\bar{v},i}}} \right)^2$$

When VDI mode is selected, the covariance analysis is always performed but only for suspect measurements for which penalties are biggest than $(1.96)^2$ according to the VDI 2048 specifications (formula 141).

Another main difference is that VALI approach is making connection between the measured values and the resulting validated values (measured and unmeasured) considering all of them as functions resulting from linear combination of measurement values when solution is reached. In the VDI2048, all the mathematical developments are made considering the deviations (n) following detailed statistical developments.

The table below details the difference in notation between VALI formulation and VDI formulation.

VALI	VDI 2048	
$\bar{y}$	x	Measured value
Y	$\bar{x}$	Validated measured variables values
		Calculated unmeasured variables values
σ^2	σ^2	Variance
σ	σ	Standard deviation
V	V	Confidence interval
F(X,Y)	f	Auxiliary conditions
A	F	Jacobian matrix of auxiliary conditions with respect to measured variables
B		Jacobian matrix of auxiliary conditions with respect to unmeasured variables
C		Constant term for constraints
P		Weighting matrix of objective function
S	S	Empirical covariance matrix
ρ	ρ	Correlation coefficient
L	ξ	Lagrange function
λ	k	Lagrange multipliers
J		Jacobian matrix of the Lagrangian system
E		Variable array of the Lagrangian system
D		Residues array of the Lagrangian system
ν	ν	Improvement
g		Result value
s		Sampled value
r		Number of redundancy

On the other side a number of abbreviations are used in VALI report files. Their correspondence to VDI nomenclature is described in the following table:

Abbreviation	Meaning (in English)	Meaning (in German)
95%Conf	95 % confidence interval	95 %-Konfidenzintervall
Abs.Acc	absolute accuracy	Absolute Messunsicherheit (hier: gemäß 95 %-Konfidenzintervall)
COR.FACT	correlation factor	empirischer Korrelationskoeffizient
COV.FACT.	covariance factor	Kovarianzfaktor, entspricht Korrelationskoeffizient nach VDI 2048 Gleichung (129) in Prozent
CST	Constant	konstant
MASSF	Massflow	Massenfluß
MEA.ACC	measurement accuracy	Messunsicherheit (hier: gemäß 95 %-Konfidenzintervall)
MEA.VAL	measurement value	Messwert
PENAL.	Penalty	Hinweis auf grobe Fehler
P.U.	physical unit	Physikalische Einheit
REC.ACC	reconciled accuracy	Validierte Messunsicherheit (hier: gemäß 95 %-Konfidenzintervall)
REC.VAL	reconciled value	Ergebniswert
Rel.Acc	relative accuracy	Relative Messunsicherheit (hier: gemäß 95 %-Konfidenzintervall)
Rel.Corr	relative corrections	Relative Verbesserungen in Prozent entspricht dem Quotienten aus der absoluten Verbesserung und dem validierten Wert
Rel.Gain	relative gain	Relativer Gewinn

SECTION 6: TRICKS AND TIPS

6.1 HOW TO INITIALIZE A MODEL ?

As VALI is based on an equation solver approach, an initial value must be set to all its variables. Adequate initial values are taken by VALI from the results of preceding runs and from user given initial values. If such a value is not available, then default values are given to the variables. However, in some cases these initial values are not adequate to ensure model convergence.

VALI VARIABLES INITIALISATION PROCEDURES

VALI has two default procedures to generate the starting values of the variables. You can change the behaviour of this procedure thanks to the parameter INIT of the OPTVAL unit:

- INIT=1: Priority is given to the measurements. Whenever it is possible, measured values will be used to determine the initial values. For unmeasured variables which cannot be determined on basis of other measurements, the initial variables values are kept if they are feasible. Otherwise, arbitrary values are used
- INIT=0: The initial variables values are kept if they are feasible. Otherwise, arbitrary values are used

By default INIT=1.

The arbitrary values are:

- T = 500 K for the temperature,
- P = 10 bar for the pressure,
- MIXF = 1 / number of outlet STREAMs in the case of a SPLVAL UNIT,
- Partial flowrates = 1 / number of Compounds kmol/s.

The default procedure based on the measurements is recommended when creating a new model. When the model has been run successfully, using the initial variables values is usually better.

METHODOLOGY FOR AN EASY MODEL INITIALISATION

An efficient way to initialise a model is to proceed as follows :

- validate each PFD separately, using only mass balances (set MODE to 1 or use the -MBO runtime option). This will usually require the addition of some pseudo-measurements that you can remove later
- validate each PFD separately, adding progressively the other balances (heat balances, liquid/vapour equilibrium, efficiencies, etc.)
- validate progressively the full model by progressively activating the various PFDs

Attention: Part of a model can be set OUT by setting units or PFDs OUT via the measurements files.

Reasonable starting points must be provided for the composition of the liquid and vapour streams

that are outlets of liquid/vapour separators (LVEVAL).

6.2 SOLVING CONVERGENCE PROBLEMS

REASONS FOR NO CONVERGENCE

After a successful verification and analysis phase, a failure to solve the model can happen in various ways:

- the Jacobian matrix can be structurally singular
- the Jacobian matrix can be numerically singular
- the model cannot be converged after the maximum number of iterations
- a thermodynamic calculation fails

In case of a **structurally singular** matrix Vali can detect what equations are introducing the singularity. These equations are listed in the file <model>.r1v.

When reaching a **numerically singular** matrix, Vali starts a Post-Mortem Analysis Phase, which sometimes, but not always, allows to detect the equations and variables linked to the problem. Again, the information will be provided in the file <model>.r1v.

When not reaching convergence after the specified **maximum number of iterations**, Vali will anyway generate a report with the best solution found so far. If at least one feasible solution (that is a solution that satisfies the constraints) has been found, then the "best solution" will be the feasible solution that minimises the total penalty. If no feasible solution has been found, then the "best solution" will be the solution that minimises the sum of residues of the constraints.

Failing to converge the validation model is in most cases due either to a wrong problem definition or to a bad initialisation point. Applying the method proposed in the section "How to initialise a model" here above should solve most of the problems due to a wrong initialisation point.

LOOKING IN VALI REPORT FILES

A first step is to look for more information into the various results file of Vali.

The file <model>.r1v will indicate, for example, inadequate initial values which are replaced by default values. You should check whether these default values are meaningful. You can look after the variables that have violated their bounds several times (search for string VIOL). These variables are often (but not always) sources of convergence problems.

You can also look in the full report (<model>.r4v) for measurements that are extensively corrected by Vali.

You can also request the generation of debugging files (-dbg runtime option), with a trace of the equation residues, variables values and measurements values through the iterations.

6.3 HOW TO CHECK THE COHERENCE OF A MODEL ?

To check the coherence of a model, you can:

- review the inlets and outlets of streams of your model in the global balance table of the report file (<model>.r4v or htm)
- look in the table of unmeasured variables and check their a posteriori accuracy. Variables with a very high a posteriori accuracy are a source of convergence problems. Additional measurements should be defined in order to reduce this uncertainty. Candidates for new measurements can be found by looking at the contributors listed in the sensitivity analysis report
- look at the constants of your model and verify whether these variables are really to be considered constant

6.4 HOW TO DEAL WITH CLOSED LOOPS ?

The modelling of a closed loop, without any purge stream leads to a singular Jacobian matrix because of inappropriate mass balances that are generated when closing loop. This problem can be solved in two ways :

- the preferred method is to open the loop and to add temperature and pressure equalities between the two corresponding streams (use unit CUTVAL)
- another solution is to add a purge and an inlet stream to the loop and to fix one of these two flow rates to a small value

6.5 DEALING WITH NEGATIVE FLOW RATES

By default, VALI does accept to reach a solution with negative flowrates when the corresponding streams are not involved in Heat balances or LVE constraints. Getting negative flow rates as a solution is usually the indication of an incoherence in the structure of your model or of major errors in your measurements.

Setting the parameter NEGATFLR of OPTVAL to 1, will avoid getting negative flow rates in any case. However, this can sometimes also prohibit convergence of the model.

Note also then when using the new SQP-IP solver, negative flow rates are never accepted.

6.6 HOW TO SWITCH BETWEEN OPERATING POINTS?

In some cases a given process can be operated in very different modes. For example, a power plant can be operated at full load or at partial load, in summer or winter conditions. The same validation model can normally be used for these various operating modes. When necessary, parts of the model can be switched off or not depending on the operation mode.

When used in on-line operation, the variations between two operating points are usually small and do not cause any convergence problem.

However, when used in off-line mode, the difference between two consecutive points can be very important and can cause convergence problems. In such a case, it is recommended to save typical operating conditions of the various modes using the DATA UNIT and DATA STREAM keywords in an output section of a measurements file. You can then re-use those typical operating conditions files as new starting point for a corresponding mode.

6.7 CUSTOMIZING REPORTS

The following topics are covered in this sub-section:

- modifying the <model>.r4v report
- requesting the <model>.prn report
- producing reports with data and measurements files

The preferred tool to build customized reports is [ValiReport](#).

CUSTOMIZING THE R4V REPORT

The REPVAL unit can be used to change the lay-out of the r4v report created by VALI.

Each section can be activated or de-activated independently and the section [_REPVAL](#) for more information.

REQUESTING THE <MODEL>.PRN REPORT

The <model>.prn report is activated via the -PRN Vali Runtime option. This report contains all the data of all streams and is designed to be imported in an Excel (or any other) work-sheet.

PRODUCING REPORTS WITH MEASUREMENT FILES

The TABLE commands of the Output section of the measurement files can be used to generate various customised reports.

Alternatively measurement files can be used to extract data related to unit and streams on a "one value per line" basis. These reports can be used as input of a spreadsheet to produce dedicated reports.

Prepare the output section of your data file to extract the required information. Then, insert the result file in your spreadsheet. Define in another section of your spreadsheet the report as required by referencing the values in the section when you imported the data file.

For a detailed description of measurements files see section "[Using measurement files](#)".

6.8 USER-DEFINED PHYSICAL UNITS

You can easily add your own physical units by introducing them in the file pud.cfg in the \usr directory of your installation. Each line defines a new physical on basis of the following syntax:

UT CF1 CF2 D NAME

where:

- UT: the unit type (see table below for units types and default Belsim units)
- CF1: a multiplying correction factor
- CF2: an adding correction factor
- D: a the number of digits after the decimal point in the result boxes
- NAME: a name of the physical units.(This name cannot contain any blank character)

CF1 and CF2 must be such that when multiplying the value expressed in user-defined units by CF1 and adding the term CF2 one gets the value expressed in Belsim default physical units.

$$\text{Bls\_Value} = \text{CF1} * \text{User\_Value} + \text{CF2}$$

For example, the definition of °F is as follows:

TE 5.5555555556E-01 2.5537222222E+02 2 F

AD	-	adimensional	MS	kJ/kmol/C	molar heat capacity
AR	m2	area	MT	kmol/m2/s	molar diffusion coefficient
CO	kW/m/K	conductivity	MV	m3/kmol	molar volume
CT	kW/K	heat transfer factor	MW	kg/kmol	molecular weight
DE	d60/60	density at 60°F vs water at 60°F	PC	Euro	currency unit
DF	m2/s	diffusivity coef./surface velocity	PO	kW	power
DP	bar	pressure difference	PR	bar	pressure
DT	K/m	temperature gradient	RE	m2*s*K/kJ	thermal resistance
FX	-	user parameters	RS	RPM	rotation speed
GL	1/m	1/length	SP	m/s	velocity
HC	kJ/K	heat capacity	SW	kg/m3	specific weight
HT	kW/m2/K	heat transfer coefficient	TC	kW/C	conduction coefficient
HW	kJ/kg	weight enthalpy	TE	K	temperature

IT	1/s	1/time	TI	s	time
K1	kmol cata/ m ³	mol of catalyst per volume	TR	K	relative temperature
K2	kg cata/m ³	weight of catalyst per volume	VF	m ³ /s	volumetric flow
LG	m	length	VO	m ³	volume
MF	kmol/s	molar flow rate	WF	kg/s	mass flow
MH	kcal/mol	molar enthalpy	WS	kJ/kg/K	heat capacity on mass basis
MM	kg	mass	WV	m ³ /kg	mass volume

Attention:

1. The names of physical units should be unique, even for different types of physical units.
2. The physical unit configuration file is read when starting ValiModeller application. After any change in this file, you must therefore restart ValiModeller application.
3. User physical units can also be defined through ValiModeller.

SECTION 7: VALI ERROR MESSAGES

7.1 VALI ERROR HANDLING FACILITY

VALI error handling facility has been developed in order to give the most relevant information on the correctness of your model and to help you in case of problems.

VALI error messages are identified by an alphanumeric code of the following form:

COD NUM MESSAGE

where:

- COD: the level of gravity of the error:
 - INF: Information: this is a simple information message
 - WRN: Warning: a particular situation that could be a source of error in the model is detected. A corrective measure has been taken, but it should not affect the quality of the results
 - ERR: Error: an error has been detected. A corrective measure has been taken. Vali should run correctly but the results could be affected
 - ERS: Severe Error: a severe error has been detected. A corrective action has been taken. Calculation will proceed but the results are probably wrong
 - ERF: Fatal Error: a fatal error has been detected for which no corrective action could be taken. The calculation will stop
- NUM: the error number (4digits)
- MESSAGE: the error that has been detected

Example:

```
*INF*2472      MEASUREMENT T STREAM FUME.1 SHOWS A
                STANDARD DEVIATION > 1.E+8 AND IS NOT TAKEN INTO ACCOUNT.
```

A summary of the errors detected in each phase is produced in the Run VALI windows and at the end of the Log file (<model>.r1v).

	INF	WRN	ERR	ERS	ERF
1 Files Reading	0	0	0	0	0
2 Verification	0	0	0	0	0
3 Analysis	14	0	1	0	0
4 Resolution	0	0	0	0	0
5 Report	0	0	0	0	0
6 Files Writing	0	0	0	0	0

FIGURE 6 - SUMMARY OF THE ERRORS

ERF 1003 NUMBER OF TAGS EXCEEDS I

You have defined too many tags. Please contact Belsim S.A.

***ERF* 2436 SINGULAR SUBSET OF EQUATIONS**

This fatal error occurs when a subset of equations contains one or more equations which are linear combinations of the other equations.

As the equations generated by VALI for each UNIT cannot be linearly dependant, the origin of this fatal error lies in the topology of the model.

A typical case is discussed below.

The process includes a loop whose material flow contains a chemical Compound which does not enter or exit this loop. To overcome the fatal error generated by this situation, remove the Compound from the THERMOD used in the STREAMS which constitutes the loop, or introduce this Compound in the THERMOD of the incoming and outgoing STREAMS.

In the <model>.r1v file, you will find:

- the list of the equations belonging to the singular subset of equations
- the list of variables (whether measured or not) as well as the constants involved in these equations

***ERF* 2512 RATES MEASUREMENT TYPE OBJECT PARAMETER HAS NO MEANING BECAUSE THE RATES CONTAINS ONLY ONE NON ZERO COMPOUND**

Due to the model definition, VALI has detected that all but one partial flowrate of this STREAM are equal to zero. Therefore, any measurement of composition has no meaning and VALI cannot consider it.

For example, a BBXVAL UNIT could have two inlet STREAMS whose associated THERMOD contains one Compound CH1 and one outlet STREAM whose associated THERMOD contains two Compounds CH1 and CH2. During the model analysis, VALI detects that CH2 cannot exist at the outlet and generates a measurement which specifies that the partial flowrate of CH2 equals zero with a zero accuracy for the outlet STREAM. The outlet stream has therefore only one component and any composition measurement has no meaning. In this case, you must introduce the Compound CH2 in the THERMOD associated to the inlet STREAMS.

Another situation similar to the first one is when the THERMOD associated to the inlet STREAMS contains also CH2 and you have specified that the inlet partial flowrates equal zero with a zero accuracy. VALI will again identify that the outlet stream has only one component and that any composition has no meaning. In this case, you must remove some CH1 composition measurements.

In general, to overcome this error, you must redefine correctly your measurements or introduce the Compound in some THERMODS.

***ERF* 2523 MEASUREMENT ... CANNOT BE EQUAL TO ZERO WITH A ZERO STANDARD DEVIATION**

This fatal error occurs when you request that a total flowrate should be equal to zero.

To overcome this error, replace the measurement value by a 1.D-6, for instance, while keeping its standard deviation equal to zero or preferably set the STREAM OFF.

***ERF* 2553 UNDETERMINED RECYCLE**

This fatal error occurs if VALI identifies a loop whose flowing stream contains at least one chemical compound for which there is no associated measurement to calculate its recycle flowrate.

Undetermined pressure loops can also occur.

7.2 ERRORS PRODUCED IN THE "MODEL ANALYSIS" PHASE

These fatal errors are due to a poor definition of your model and consist mainly in:

- a lack of measurements
- wrong measurement values

Overcoming these fatal errors generally requires to look for more details in the file <model>.r1v file.

***ERF* 2424 NOT ENOUGH MEASUREMENTS TO SOLVE WHOLE PROBLEM**

This error occurs when there are not enough measurements to reach a positive level of redundancy in all parts of the model. This error can occur even when the global level of redundancy is positive if there are not enough measurements in a sub-section of the process.

In such a case a list of supplementary measurements is produced in the <model>.r1v file, together with the list of variables and equations that can be validated or at least calculated.

There are three ways to overcome this situation:

- add new measurements
- simplify the description of the process
- add new equations

Add new measurements

You can directly cut and paste some of the TAG definitions proposed by VALI in the <model>.r1v file.. Select one TAG definition in each group of supplementary measurements

*ERF*2424 NOT ENOUGH MEASUREMENTS TO SOLVE WHOLE PROBLEM					
SUPPLEMENTARY MEASUREMENT (1) OUGHT TO BE CHOSEN AMONG VARIABLES:					
POWT_POWER	STREAM	POWT	POWER	31710.	-5.0 kW
COMP_EFFIC	UNIT	COMP	EFFIC	0.76000	-5.0 -
POWC_POWER	STREAM	POWC	POWER	15610.	-5.0 kW
FUME.2_T	STREAM	FUME.2	T	1009.3	-5.0 C
AIR.1_T	STREAM	AIR.1	T	15.000	-5.0 C
AIR.2_P	STREAM	AIR.2	P	14.932	-5.0 bar
FUME.1_P	STREAM	FUME.1	P	14.832	-5.0 bar
SUPPLEMENTARY MEASUREMENT (2) OUGHT TO BE CHOSEN AMONG VARIABLES:					
POWT_POWER	STREAM	POWT	POWER	31710.	-5.0 kW
COMP_EFFIC	UNIT	COMP	EFFIC	0.76000	-5.0 -
POWC_POWER	STREAM	POWC	POWER	15610.	-5.0 kW
FUME.2_T	STREAM	FUME.2	T	1009.3	-5.0 C
AIR.1_T	STREAM	AIR.1	T	15.000	-5.0 C
AIR.2_T	STREAM	AIR.2	T	432.80	-5.0 C
FUME.1_T	STREAM	FUME.1	T	1647.5	-5.0 C

FIGURE 7 - LIST OF SUPPLEMENTARY MEASUREMENTS TO BE SELECTED

Simplify the description of the process

To help you, VALI lists the equations which cannot be considered during data validation.

The names of the equations which are in this list gives you an idea on the UNITS involved in at least one process subsection which suffers from lack of information. A careful analysis of these UNITS may sometimes indicate the possibility of grouping several UNITS into a single one.

Add new equations

If the variables and the equations which appear in the lists are related to a liquid-vapour separator, you could add the liquid-vapour relationships. Anyway, you must be aware that the quality of the results will depend upon the ability of the thermodynamic model you are using to describe the liquid and vapour fugacities of the mixture.

It is also possible to introduce your own modelling constraints with OPXVAL or FLXVAL type of units.

ERF 2436 SINGULAR SUBSET OF EQUATIONS

This fatal error occurs when a subset of equations contains one or more equations which are linear combinations of the other equations.

As the equations generated by VALI for each UNIT cannot be linearly dependant, the origin of this fatal error lies in the topology of the model.

A typical case is discussed below.

The process includes a loop whose material flow contains a chemical Compound which does not enter or exit this loop. To overcome the fatal error generated by this situation, remove the Compound from the THERMOD used in the STREAMS which constitutes the loop, or introduce this Compound in the THERMOD of the incoming and outgoing STREAMS.

In the <model>.r1v file, you will find:

- the list of the equations which belong to the singular subset of equations
- the list of variables (whether measured or not) as well as the constants involved in these equations

***ERF* 2483 UNFEASIBLE MEASUREMENT VALUES**

This fatal error occurs when some of the measurement values are not physically possible.

Typical situations are:

- a temperature (K) lower than 0
- a composition (-) higher than 1
- a composition or flowrate lower than 0
- a composition or flowrate equal to 0 with an accuracy which differs from 0

***ERF* 2489 OVERSPECIFICATIONS**

In the list of measurements, you set the accuracy to zero of some of them to consider them constants. The error occurs when the number of constants is so high that the number of equations in which they occur is bigger than the number of involved variables. The solution of these equations is then impossible.

***ERF* 2490 THE VARIABLE ... DOES NOT OCCUR IN THE EQUATIONS**

This error will be issued for all pressure variables that are only appearing in heat balances equations where their influence is assumed to be negligible in the analysis phase.

This fatal error could also appear if you are using a new type of UNIT. In that case, contact BELSIM S.A to overcome the error.

***ERF* 2491 THE EQUATION ... DOES NOT CONTAIN ANY VARIABLE**

This fatal error could only appear if you are using a new type of UNIT. Contact BELSIM S.A to solve this error.

***ERF* 2493 PARTIAL FLOWRATE MEASUREMENT ... CANNOT BE CONSIDERED BECAUSE VALI HAS DETECTED THAT IT IS EQUAL TO ZERO**

This fatal error occurs when you provide a partial flowrate measurement while VALI detected that this partial flowrate should be equal to zero.

A typical situation refers to a mixer whose outlet STREAM contains a Compound which is not present in the inlet STREAMS. In this case, VALI generates a measurement which specifies that the related partial flowrate is equal to zero with an accuracy equal to zero. The measurement you

provide enters into conflict with the measurement generated by VALI.

To overcome this error, you must redefine correctly the THERMODs involved in the model.

***ERF* 2506 COMPOSITION MEASUREMENT ... CANNOT BE CONSIDERED BECAUSE VALI HAS DETECTED THAT THE RELATED PARTIAL FLOWRATE IS EQUAL TO ZERO**

This fatal error occurs when you provide a composition measurement with a non zero accuracy while VALI has detected that the associated partial flowrate is equal to zero.

A typical situation refers to a mixer where you have measured at the outlet a significant composition for one Compound which is not present in the THERMOD associated with the inlet STREAMS.

VALI will, in this case, eliminate the mass balance equation and add a measurement which specifies that this outlet partial flowrate is equal to zero with an accuracy equal to zero.

To overcome this error, introduce the Compound in the appropriate THERMOD in order to make your measurement consistent with your model definition.

***ERF* 2512 RATES MEASUREMENT ... HAS NO MEANING BECAUSE THIS RATES CONTAINS ONLY ONE NON ZERO COMPOUND**

Due to the model definition, VALI has detected that all but one partial flowrates of this STREAM are equal to zero. Therefore, any measurement of composition has no meaning and VALI cannot consider it.

For example, a BBXVAL UNIT could have two inlet STREAMs whose associated THERMOD contains one Compound CH1 and one outlet STREAM whose associated THERMOD contains two Compounds CH1 and CH2. During the model analysis, VALI detects that CH2 cannot exist at the outlet and generates a measurement which specifies that the partial flowrate of CH2 equals zero with a zero accuracy for the outlet STREAM. The outlet stream has therefore only one component and any composition measurement has no meaning. In this case, you must introduce the Compound CH2 in the THERMOD associated to the inlet STREAMs.

Another situation similar to the first one is when the THERMOD associated to the inlet STREAMs contains also CH2 and you have specified that the inlet partial flowrates equal zero with a zero accuracy. VALI will again identify that the outlet stream has only one component and that any composition has no meaning. In this case, you must remove some CH1 composition measurements.

In general, to overcome this error, you must redefine correctly your measurements or introduce the Compound in some THERMODs.

***ERF* 2523 MEASUREMENT ... CANNOT BE EQUAL TO ZERO WITH A ZERO STANDARD DEVIATION**

This fatal error occurs when you request that a total flowrate should be equal to zero.

To overcome this error, replace the measurement value by a 1.D-6, for instance, while keeping its standard deviation equal to zero or preferably set the STREAM OFF.

***ERF* 2535 ALL THE VARIABLES OCCURRING IN EQUATION ARE MEASUREMENTS WITH A ZERO STANDARD DEVIATION**

This fatal error occurs when all the variables occurring in the specified equation are considered as constants because of measurements with a standard deviation equal to zero.

To overcome this error, identify the occurring variables and provide a non zero value for the standard deviation of one of them.

***ERF* 2553 UNDETERMINED RECYCLE**

This fatal error occurs if VALI identifies a loop whose flowing stream contains at least one chemical compound for which there is no associated measurement to calculate its recycle flow rate.

Undetermined pressure loops can also occur.

7.3 ERRORS PRODUCED IN THE "MODEL RESOLUTION" PHASE

As VALI works as an equation solver, it is often necessary to examine the residuals of each equations at each iteration as well as the values of the occurring variables.

These results can be looked at in the <model>.r3v file or in the trace files produced through the -DBG runtime option.

Whether you run your model for the first time or not, VALI automatically generates starting values for all the variables. No convergence situations may of course be due to poor estimates of these variables. The procedure used in VALI is explained in section "[How to initialize a model](#)".

***ERF* 2501 MAXIMUM CAPACITY OF JACOBIAN MATRIX IS REACHED**

This error occurs when the maximum number of non zero elements in the Jacobian matrix is reached.

A way to overcome this situation is to split your model into smaller models that can be run successively.

If you cannot proceed in this way, contact BELSIM S.A to get a copy of VALI with a larger size for the arrays. The maximum size depends upon the size of core memory of your computer.

***ERF* 2509 ALL PARTIAL FLOWRATES OF RATES ... BECOME EQUAL TO ZERO**

This error occurs when all the partial flowrates of a STREAM become equal to zero.

In most cases, this STREAM corresponds to a process purge and the flowrates and composition measurements which have been provided are of poor quality.

To overcome this error, we advise you to specify a small value for the incriminated flowrate together with a zero accuracy and to run again VALI. The analysis of the measurement correction should help you to identify the poor measurements.

***ERF* 2540 THE JACOBIAN MATRIX IS SINGULAR**

This is certainly one of the most difficult error to overcome.

Two common situations may generate this fatal error.

The first one concerns the case where you have measured the temperature of a one component stream which is in the liquid-vapour state.

The second one deals with recycling streams where you do not provide any measurement to fix their flowrate.

We advise you to decompose your model into smaller ones and to run them separately in order to identify the process section which causes the error. It will then be easier to discover its origin.

***ERF* 2551 NO CONVERGENCE DURING THERMODYNAMIC STATE CALCULATION OF STREAM ... FOR T ... AND P ...**

This error occurs when the state calculation, of the given STREAM for the given temperature and pressure, do not converge.

This can be due to the fact that the starting values for these variables are poor. We advise you to provide better ones as explained in section "[How to initialize a model](#)".

A second way to overcome this error is to provide realistic STREAM states for each stream in order to simplify the calculations.

***ERF* 2552 NO CONVERGENCE DURING THERMODYNAMIC STATE CALCULATION OF STREAM ... FOR H ... AND P ...**

This error occurs when the state calculation of the given STREAM for the given enthalpy and pressure do not converge. The related STREAM contains only one Compound so that the basic variable is the molar enthalpy and not the temperature.

This can be due to the fact that the starting values for these variables are poor.

We advise you to provide better ones as explained in section "[How to initialize a model](#)".

A second way to overcome this error is to provide realistic STREAM states for each stream in order to simplify the calculations.

***ERF* 2303 THERMODYNAMIC INCOHERENCE FOR THE ZONE - MPL AND MPV EQUAL : INSTEAD OF 0 ET 1**

A critical review of the results indicates that the outlet streams of a unit involving phase equilibrium are not really in equilibrium.

This can occur when the trivial solution (same composition in all phases) is obtained.

A better starting point must be given by initializing the compositions of the outlet streams to more

realistic values.

This error can also occur when the streams are in super-critical state, in which case Liquid/Vapour phase separation is meaningless.

***ERF* 2552 NO CONVERGENCE DURING THERMODYNAMIC STATE CALCULATION OF STREAM NAME. FOR H ... AND P ...**

This error occurs when the state calculation of the given STREAM for the given enthalpy and pressure do not converge. The related STREAM contains only one Compound so that the basic variable is the molar enthalpy and not the temperature.

This can be due to the fact that the starting values for these variables are poor.

We advise you to provide better ones as explained in section "[How to initialize a model](#)".

A second way to overcome this error is to provide realistic STREAM states for each stream in order to simplify the calculations.



belsim

CHAPTER 6

DATA VALIDATION UNITS

SECTION 1: SELECTING UNIT TYPES

The type of a UNIT determines what type of equipment is modeled and what equations are generated.

- Black-box/Mixer:
 - BBXVAL: mixes up to 10 Streams and splits them into up to 10 Streams of different compositions
 - EFFVAL: mechanical/thermal Stream efficiency box
 - CUTVAL: allows breaking the material or pressure loops that can appear when dealing with closed cycles
 - NETVAL: network of pipes where Streams are mixed and split up on basis of a product code
- Column:
 - COLVAL: saturation state of outlet Streams (saturated liquid or vapor) can be specified
 - DISVAL: tray by tray distillation column calculation model
 - CRUVAL: same as BBXVAL but petroleum fractions can be identified
- Compressor:
 - COMVAL: includes isentropic efficiency equation
- Controllers (user-equations):
 - ATMOPRES: specifies atmospheric pressure (used a reference for pressure gauges)
 - EQUVAL: equality of composition between 2 streams
 - FLEXLIB: user-defined functions
 - FLXVAL: user-defined equations
 - OPXVAL: simple equations involving 2 variables
 - VARVAL: additional variables
- Heat exchanger:
 - ECOVAL: detailed simulation of a heat exchange section in an HRSG.
 - EEXVAL: can model any heat exchange, no performance equation
 - HEXVAL: heat exchange between two Streams with performance equation
 - PRHVAL: heat exchange between a vapor that is condensed and a cold stream
 - SATVAL: allows you to set any saturation state on a Stream
 - WALVAL: estimation of heat losses through walls of vessels
- Pump:

- PUMVAL: useful if one wants to identify pump power or if its power is known or related to another equipment (driven by a turbine). In many case, pumps are not modeled.
- Reactor:
 - BATVAL: any Reaction can occur (atomic mass balance)
 - BWHVAL: as BATVAL but with chemical equilibrium
 - CND-HNO₃: Nitrous Gas Condenser
 - HEX-HNO₃: Nitrous Gas Cooler
 - RCT-HNO₃: NH₃ Oxidation
 - RCTVAL: the possible Reactions are specified
 - RKIVAL: as RCTVAL but based on kinetics
 - RKPVAL: as RCTVAL but some reactions can be at (some distance of) chemical equilibrium
 - RLVVAL: as RCTVAL but with 2 outlet Streams in L/V equilibrium
- Separator:
 - LLEVAL: Liquid/Liquid equilibrium
 - LLVVAL: Liquid/Liquid/Vapor equilibrium
 - LVEVAL: Liquid/Vapor equilibrium
 - WLEVAL: Water/Liquid equilibrium
 - WLVVAL: Water/Liquid/Vapor equilibrium
- Splitter:
 - SPLVAL: splits a Stream into up to 10 Streams of same composition
- Storage:
 - TNKVAL: material balance of storage tank (involves accumulation and de-accumulation)
- Turbine:
 - TURVAL: includes isentropic efficiency equation
- Valves:
 - DPVAL: pressure drop in valves (can be function of rate)
 - PIPEVAL: models the pressure drop along pipes

MATHEMATICAL UNITS

Mathematical units can only be included in the Main PFD. They determine various global settings at the model level.

- BNDVAL: overwrites the default bounds set on variables.
- CRUOPT: defines options used in petroleum fractions, TBP and ATSM handling
- CUTOFF: defines the threshold values for the generic CUTOFF on various types of measurements
- GROSSERR: specifies the settings of the Gross Error Elimination algorithm
- OPTVAL: defines various mathematical and reporting settings
- PRECIS: defines the precision criteria use for thermodynamic calculations
- REPVAL: controls the amount and the type of information printed out in the main report file (r4v)

SECTION 2: ATMOPRES: SETTING ATMOSPHERIC PRESSURE

PURPOSE

The **ATMOPRES** unit allows you to introduce one or several measurements of the atmospheric pressure. All pressure gauges expressed on a relative basis will be converted to absolute pressure according to the so defined atmospheric pressure.

If several measurements are defined on the atmospheric pressure, the actual reference pressure will be set to an weighted average value according to the respective uncertainties of the measurements:

$$Atmo\_pres = \frac{\sum_i \frac{MEA_i}{\sigma_i^2}}{\sum_i \frac{1}{\sigma_i^2}}$$

where:

- atmo_pres: the atmospheric pressure
- MEAi: the atmospheric pressure measurements
- σ_i : the relative standard deviation. If the measurement accuracy is absolute, relative accuracy is recalculated

Attention: Only one ATMOPRES unit can appear in model file. Its name should start with the letters ATMOP and no other unit should have a name starting by ATMOP.

SYMBOLS



FIGURE 1 - SYMBOLS FOR UNIT ATMOPRES

CONNECTION RULES

No connections are allowed on the **ATMOPRES** unit.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
PATMO	Parameters	Measurable	Atmospheric Pressure

SECTION 3: BATVAL: REACTOR WITH ATOMIC BALANCES

PURPOSE

The **BATVAL** unit allows you to model reactor and takes the atomic balances, energy balance and pressure drop into account.

It is well suited for reactors with unknown reaction stoichiometry.

SYMBOLS

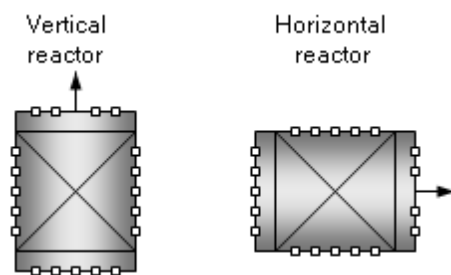


FIGURE 2 - SYMBOLS FOR UNIT BATVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 10
- Type: material (at least one), thermal, mechanical

OUTLET STREAMS

- Maximum number: 10
- Type: one material as the first outlet (as shown on the symbols), thermal, mechanical

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
COMPMODE	General	Input	Compound balance mode
MODEDP	General	Input	Pressure equation

NAME	TAB	TYPE	DESCRIPTION
DP	Parameters	Measurable	Pressure drop
DPX	Parameters	Input	Nominal pressure drop
MASSFx	Parameters	Input	Nominal flow rate
CFDP	Parameters	Measurable	DP correction factor
REFDP	Parameters	Input	Reference stream for pressure drop

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	Atomic and Energy Balance	Atomic and energy balances are taken into account
1	Atomic Balance only	Only atomic balances are taken into account

COMPMODE (General)

Compound balance mode (type: Input).

1	Enabled	Atomic balances are taken into account
3	None	Only global mass balance (without atomic balances) is taken into account.

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form: $DP = CFDP * DPX * (MASSF / MASSFX)^2$ where: ■ CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%) ■ DPX: the nominal pressure drop ■ MASSFX: the nominal flow rate ■ MASSF: the actual flow rate

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the reference inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if $MODEDP = 0$.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

MASSFX (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when MODEDP = 2.

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

EXAMPLE : COMBUSTION OF NATURAL GAS

Attention: see model batval.bls in the directory "examples\batval". A unit of type BATVAL is also used in the example coal.bls (coal fired power plant).

A pressure drop of 0.05 bar is taken into account between the air inlet and the fumes outlet.

The atomic and energy balances are taken into account simultaneously and a heat loss is modeled.

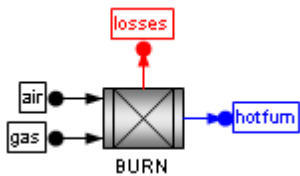


FIGURE 3 - BATVAL EXAMPLE: COMBUSTION OF NATURAL GAS

There is 1 UNIT :

- BURN:
 - type=BATVAL
 - Symbol=Horizontal Reactor
 - MODE='Atomic and Energy balance'
 - MODDP='DP variable'
 - REFDP=AIR

4 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
AIR	AIR	to BURN
GAS	GAS	to BURN
HOTFUM	FUMES	from BURN
LOSSES	THERMAL STREAM	from BURN

The Compounds associated to each THERMOD are respectively:

- GAS=C1, C2, N2
- AIR=N2, O2
- FUMES=N2, O2, H2O, CO2

The list of measurements is given in table below.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
AIR_MFN2	STREAM	AIR	MFN2	79.000	CST	%
AIR_MFO2	STREAM	AIR	MFO2	21.000	CST	%
AIR_P	STREAM	AIR	P	1.0000	0.100	bar
AIR_T	STREAM	AIR	T	20.000	CST	C
BURN_DP	UNIT	BURN	DP	0.50000E-01	-5.000	bar
GAS_MFC2H6	STREAM	GAS	MFC2	3.0000	CST	%
GAS_MFCH4	STREAM	GAS	MFC1	90.000	CST	%
GAS_MFN2	STREAM	GAS	MFN2	7.0000	CST	%
GAS_MOLF	STREAM	GAS	MOLF	100.00	-5.000	Nm3/h
GAS_P	STREAM	GAS	P	30.000	0.100	bar
GAS_T	STREAM	GAS	T	20.000	CST	C
HOTFUM_MGO2	STREAM	HOTFUM	MGO2	2.1000	-10.00	%
HOTFUM_P	STREAM	HOTFUM	P	1.0000	0.100	bar
HOTFUM_T	STREAM	HOTFUM	T	1700.0	20.0	C
LOSSES_LOAD	STREAM	LOSSES	LOAD	200.00	-10.00	kW

TABLE 1 - MEASUREMENTS FOR BATVAL EXAMPLE

The composition of Streams GAS and AIR is assumed to be known perfectly.

The O2 content of the fumes is given on a dry basis.

The results are (Tags Report):

Calculated excluding tags off

Number of tags	15
Number of equations	14
Number of unmeasured variables	12
Number of measured variables	8
Number of variables considered constant	7
Obvious number of redundancies	2
Total number of redundancies	2
Number of trivial redundancies	0

Validation and reconciliation results

Objective function	12.9409
Chi-square test value	5.99000
Sum of squares of residues	0.308954E-011
Number of active bounds	0
Number of variables close to bounds	0
Number of eliminated tags	0
Number of relaxed tags	0
Number of equivalencies	0
Number of flagged tags	1
Number of filtered tags	0
Tags out of range	0

TAGS IN THE MODEL

Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit	Comment
AIR_MFN2	79.000	CST	79.000			%	
AIR_MFO2	21.000	CST	21.000			%	
AIR_P	1.0000	0.100	1.0250	0.707E-01	0.06	bar	
AIR_T	20.000	CST	20.000			C	
BURN_DP	0.50000E-01	5.00 %	0.49984E-01	5.00 %	0.00	bar	
GAS_MFC2H6	3.0000	CST	3.0000			%	
GAS_MFCH4	90.000	CST	90.000			%	
GAS_MFN2	7.0000	CST	7.0000			%	
GAS_MOLF	100.00	5.00 %	104.86	4.59 %	0.95	Nm3/h	
GAS_P	30.000	0.100	30.000	0.100	0.00	bar	
GAS_T	20.000	CST	20.000			C	
HOTFUM_MGO2	2.1000	10.0 %	1.8882	10.7 %	1.02	%	
HOTFUM_P	1.0000	0.100	0.97501	0.707E-01	0.06	bar	
HOTFUM_T	1700.0	20.0	1667.1	17.8	2.71	C	
LOSSES_LOAD	* 200.00	10.0 %	142.92	8.45 %	8.14	kW	

TABLE 2 - RESULTS OF BATVAL EXAMPLE

SECTION 4: BBXVAL: BLACK BOX WITH MOLAR BALANCES

PURPOSE

This type of UNIT allows you to consider mass balances, energy balance and pressure drops for any process unit where no chemical reaction occurs. It is used to model mixers, distillation, absorption and stripping towers and process sections that are globally taken into account. It acts like a black box satisfying the balance equations.

The **BBXVAL** type can also be used to mix or split thermal and mechanical streams, for example to mix (add) the power produced by several turbines on the same shaft.

SYMBOLS

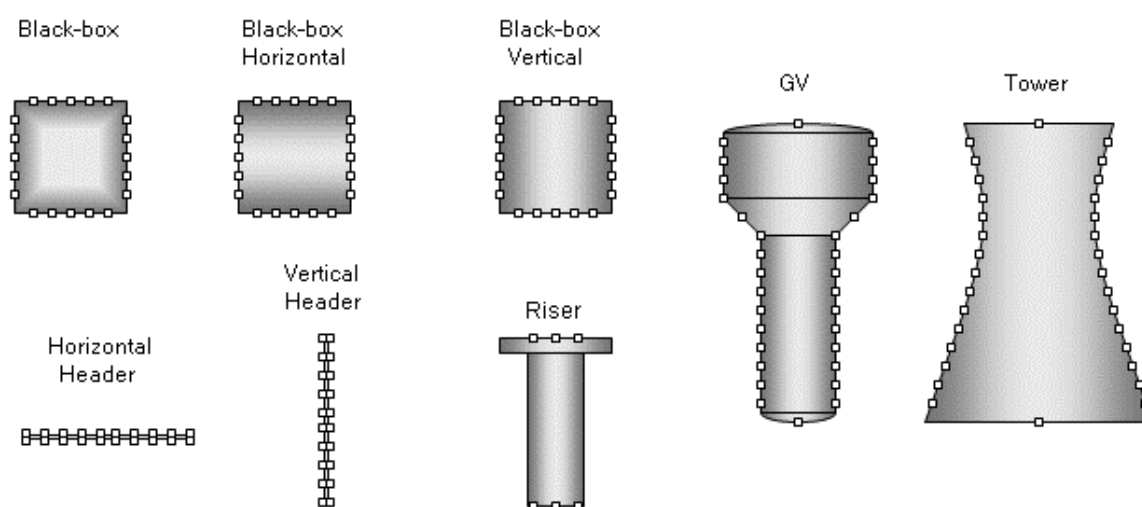


FIGURE 4 - SYMBOLS FOR UNIT BBXVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 10
- Type: material, thermal, mechanical

OUTLET STREAMS

- Maximum number: 10
- Type: material, thermal, mechanical

Attention: The Horizontal and Vertical headers only support Material Streams.

If one or more material Streams are connected as inlet, then at least one material Stream must be

connected at the outlet, and vice versa.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
COMPMODE	General	Input	Compound balance mode
MODEDP	Parameters	Input	Pressure equation
MODETOEQ	General	Input	Temperature equalities
DPi	Parameters	Measurable	Pressure drop
DPXi	Parameters	Input	Nominal pressure drop
MASSFXi	Parameters	Input	Nominal flow rate
CFDPi	Parameters	Measurable	DP correction factor
REFDP	Parameters	Input	Reference stream for pressure drop
FLOWFILT	General	Input	Automatic flow rate filtering

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Mass and Energy Balance	Mass and energy balances are taken into account
1	Mass Balance only	Only mass balances are taken into account

COMPMODE (General)

Compound balance mode (type: Input).

1	Enabled	Compound balances are taken into account
3	None	Only global mass balance (without compound balances) is taken into account.

MODETOEQ (General)

Temperature equalities (type: Input).

0	None	no temperature equality equation will be generated
1	between all outlet stream	all outlet streams will be set at the same temperature
2	between first inlet and outlet stream	all outlet streams and the first inlet stream will be set at the same temperature

Temperature equalities are only added when the unit is calculated in mass balance only. When the energy balance is activated, these additional equations will never be generated.

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * (MASSF / MASSFX)^2$$

where:

- CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%)
- DPX: the nominal pressure drop
- MASSFX: the nominal flow rate
- MASSF: the actual flow rate

A value of MODEDP has to be given for each outlet stream, although it is only meaningful for material streams.

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

DP is calculated independently for each stream.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when MODEDP = 2.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when MODEDP = 2.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when MODEDP = 2.

These parameters are defined independently for each stream.

FLOWFILT (Parameters)

Automatic flow rate filtering (type: Input).

0	Disabled	The automatic flow rate filtering is not activated
1	Enabled	The automatic flow rate filtering is activated

When all **but one** of the connected streams are measured, there is a risk that the unmeasured stream hits its lower bound.

This in particular the case on products routing in production accounting applications where the main production flow rate is measured and the product routing is specified though manual inputs on the flows that are redirected from the normal route (on-spec product) to slop tanks or recirculation

lines.

The automatic filtering (when enabled) creates an estimate on the unmeasured flow rate based on the measured inlet and outlet flow rates. Should the estimated flow rate fall below a given cutoff value, the unmeasured stream will be set off.

EXAMPLE : CO2 REMOVAL FROM NH3 FEED GAS

Attention: see model `bbxval.bls` in the directory "`\examples\bbxval`". Units of type **BBXVAL** are also used in the example `coal.bls` (coal fired power plant).

In this example, the **BBXVAL** type of unit is used to model the removal of CO₂ from a stream containing Nitrogen, Hydrogen and Helium. Only the mass balances are taken into account. The **FLXVAL** unit **RECOVERY** calculates the recovery of CO₂ in the vent stream.

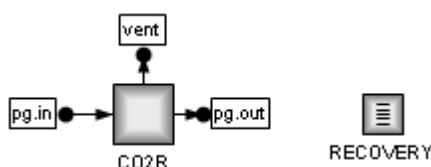


FIGURE 5 - BBXVAL EXAMPLE: CO2 REMOVAL

There are 2 Units :

- CO2R:
 - type=BBXVAL
 - MODE='Mass Balance only'
 - MODEDP1='None'
 - MODEDP2='None'
- RECOVERY:
 - type=FLXVAL
 - NEQU=1

3 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
PG.IN	PGMOD	to CO2R
PG.OUT	PGMOD	from CO2R
VENT	VENTMOD	from CO2R

The Unit **RECOVERY** uses the following Flex code:

```
FUNCTION EQ1
!
CALL GTVARTAG ('PG.IN_MOLF', pgin)
```

```

CALL GTVARTAG ('PG.IN_MFCO2', co2in)
co2in=co2in*pgin
!
CALL GTVARTAG ('VENT_MOLF', vent)
CALL GTVARTAG ('VENT_MFCO2', co2vent)
co2vent=co2vent*vent
!
RES=co2vent/co2in
END

```

The thermodynamic calculation methods are the ones used by default, as we are only concerned with the mass balances of the unit.

The compounds associated to each THERMOD are respectively:

- PGMOD= H2, N2, HE, CO2
- VENTMOD=H2, CO2

Table below gives the list of measurements.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
PG.IN_MFCO2	STREAM	PG.IN	MFCO2	39.000	-2.000	%
PG.IN_MFH2	STREAM	PG.IN	MFH2	15.000	-2.000	%
PG.IN_MFHE	STREAM	PG.IN	MFHE	1.0000	-2.000	%
PG.IN_MFN2	STREAM	PG.IN	MFN2	45.000	-2.000	%
PG.IN_MOLF	STREAM	PG.IN	MOLF	35000.	-5.000	Nm3/h
PG.OUT_MFCO2	STREAM	PG.OUT	MFCO2	0.14000	-10.00	%
RECOVERY_CO2	UNIT	RECOVERY	VAR1		OFF	%
VENT_MFCO2	STREAM	VENT	MFCO2	99.000	-2.000	%
VENT_MFH2	STREAM	VENT	MFH2	1.0000	-2.000	%
VENT_MOLF	STREAM	VENT	MOLF	14000.	-5.000	Nm3/h

TABLE 3 - MEASUREMENTS FOR BBXVAL EXAMPLE

The tag RECOVERY_CO2 is defined as a KPI (Mode CALC) with a target value of 98.8%

The results are (file r6v + section KPI of r4v):

Number of tags	10
Number of equations	14
Number of unmeasured variables	11
Number of measured variables	9
Number of variables considered constant	0
Obvious number of redundancies	3
Total number of redundancies	3
Number of trivial redundancies	0

Validation and reconciliation results

Objective function	0.580914E-001
Chi-square test value	7.81000
Sum of squares of residues	0.178034E-021
Number of active bounds	0
Number of variables close to bounds	0
Number of eliminated tags	0
Number of flagged tags	0
Number of filtered tags	0

TAGS IN THE MODEL

Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
PG.IN_MFCO2	39.000	2.00 %	39.031	1.51 %	0.00	%
PG.IN_MFH2	15.000	2.00 %	14.997	1.94 %	0.00	%
PG.IN_MFHE	1.0000	2.00 %	0.99999	2.00 %	0.00	%
PG.IN_MFN2	45.000	2.00 %	44.972	1.34 %	0.00	%
PG.IN_MOLF	35000.	5.00 %	35289.	3.61 %	0.03	Nm3/h
PG.OUT_MFCO2	0.14000	10.0 %	0.13999	10.0 %	0.00	%
RECOVERY_CO2		OFF	99.782	0.224E-01		%
VENT_MFCO2	99.000	2.00 %	99.000	0.202E-01%	0.00	%
VENT_MFH2	1.0000	2.00 %	1.0000	2.00 %	0.00	%
VENT_MOLF	14000.	5.00 %	13882.	3.62 %	0.03	Nm3/h

TABLE 4 - RESULTS OF BBXVAL EXAMPLE

SECTION 5: BNDVAL: DEFAULT BOUNDS ON VARIABLES

PURPOSE

This type of UNIT allows you to overwrite the default bounds set on variables. The bounds specified on the variables themselves through the Tags will nevertheless be used when they are more restrictive than the default bounds.

Only one unit of type **BNDVAL** can be created. It must always be created in the Main PFD.

SYMBOLS



FIGURE 6 - SYMBOL FOR UNIT BNDVAL

CONNECTION RULES

No connections are allowed on a **BNDVAL**.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	MODE (0: BNDVAL is ignored, 1: BNDVAL is used)
EFFICBND	General	Input	Default Bounds on Efficiencies (0: No, 1 : YES)
MinT	Parameters	Input	Minimum for Temperatures (default = 230 K)
MaxT	Parameters	Input	Maximum for Temperatures (default = 4000 K)
MinP	Parameters	Input	Minimum for Pressures (default = 0.01 bar)
MaxP	Parameters	Input	Maximum for Pressures (default = 1000 bar)
MinMR	Parameters	Input	Minimum for Partial Molar Rates (default = 0 kmol/s)
MaxMR	Parameters	Input	Maximum for Partial Molar Rates (default=100 000 kmol/s)

NAME	TAB	TYPE	DESCRIPTION
MinH	Parameters	Input	Minimum for Enthalpies (default = -1E+10 kJ/kmol)
MaxH	Parameters	Input	Maximum for Enthalpies (default = 1E+10 kJ/kmol)
MinLOAD	Parameters	Input	Minimum for Loads (default = -1E+7 kW)
MaxLOAD	Parameters	Input	Maximum for Loads (default = 1E+7 kW)
MinPOWER	Parameters	Input	Minimum for Powers (default = -1E+7 kW)
MaxPOWER	Parameters	Input	Maximum for Powers (default = 1E+7 kW)
MinDP	Parameters	Input	Minimum for Pressure Drops (default = 0 bar)
MaxDP	Parameters	Input	Maximum for Pressure Drops (default = 1000 bar)
MinMOLF	Parameters	Input	Minimum for Molar Flowrates (default = 0 kmol/s)
MaxMOLF	Parameters	Input	Maximum for Molar Flowrates (default = 100 000 kmol/s)
MinMASSF	Parameters	Input	Minimum for Mass Flowrates (default = 0 kg/s)
MaxMASSF	Parameters	Input	Maximum for Mass Flowrates (default = 1 000 000 kg/s)
MinMF	Parameters	Input	Minimum for Molar Fractions (default = 0)
MaxMF	Parameters	Input	Maximum for Molar Fractions (default = 1)
MinWF	Parameters	Input	Minimum for Weight Fractions (default = 0)
MaxWF	Parameters	Input	Maximum for Weight Fractions (default = 1)
MinVOLF	Parameters	Input	Minimum for Volume Flowrates (default = 0 m3/s)
MaxVOLF	Parameters	Input	Maximum for Volume Flowrates (default = 1E+5 m3/s)
MinDENS	Parameters	Input	Minimum for Densities (1E-6 D60/60)
MaxDENS	Parameters	Input	Maximum for Densities (20 D60/60)

NAME	TAB	TYPE	DESCRIPTION
MinAPI	Parameters	Input	Minimum for API Densities (0.0 API)
MaxAPI	Parameters	Input	Maximum for API Densities (default = 2000 API)
MinHUMID	Parameters	Input	Minimum for Humidity (default = 0)
MaxHUMID	Parameters	Input	Maximum for Humidity (default = 2000)
MinTBPV	Parameters	Input	Minimum for TBPV (default = 0 K)
MaxTBPV	Parameters	Input	Maximum for TBPV (default = 2000 K)
MinTBPW	Parameters	Input	Minimum for TBPW (default = 0 K)
MaxTBPW	Parameters	Input	Maximum for TBPW (default = 2000 K)
MinAREA	Parameters	Input	Minimum for Areas (default = 1E-6 m2)
MaxAREA	Parameters	Input	Maximum for Areas (default = 1E+6 m2)
MinK	Parameters	Input	Minimum for K (default = 1E-6)
MaxK	Parameters	Input	Maximum for K (default = 1E+6)
MinU	Parameters	Input	Minimum for U (default = 1E-8 kW/m2 K)
MaxU	Parameters	Input	Maximum for U (default = 1E+8 kW/m2 K)
MinDT	Parameters	Input	Minimum for DT (-2000 K)
MaxDT	Parameters	Input	Maximum for DT (2000 K)
MinDMASS	Parameters	Input	Minimum for DMass (default = -1E+8 kg), and for MASS1 (initial mass inventory of tanks)
MaxDMASS	Parameters	Input	Maximum for Dmass (default = 1E+8 kg), and for MASS2 (final mass inventory of tanks)

SECTION 6: BWHVAL: EQUILIBRATED REACTOR WITH ATOMIC BALANCES

PURPOSE

The **BWHVAL** unit allows you to model a reactor and takes the atomic balances, energy balance and pressure drop into account. The reactor can set at equilibrium or at a given distance of equilibrium.

It is well suited for reactors with unknown or complex reaction stoichiometry.

SYMBOLS

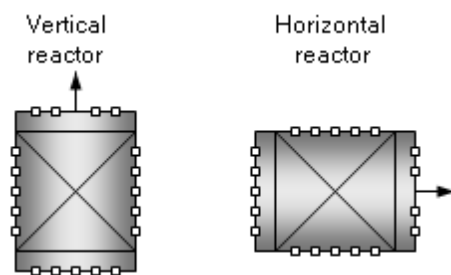


FIGURE 7 - SYMBOLS FOR UNIT BWHVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 10
- Type: material (at least one), thermal, mechanical

OUTLET STREAMS

- Maximum number: 10
- Type: one material as the first outlet (as shown on the symbols), thermal, mechanical

All material streams must use the same Thermod.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
EQ	General	Input	Selection of equilibrium equation

NAME	TAB	TYPE	DESCRIPTION
DP	Parameters	Measurable	Pressure drop
DPX	Parameters	Input	Nominal pressure drop
MASSF	Parameters	Input	Nominal flow rate
CFDP	Parameters	Measurable	DP correction factor
REFDP	Parameters	Input	Reference stream for pressure drop

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	Atomic and Energy Balance	Atomic and energy balances are taken into account
1	Atomic Balance only	Only atomic balances are taken into account

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * (MASSF / MASSFX)^2$$

where:

- CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%)
- DPX: the nominal pressure drop
- MASSFX: the nominal flow rate
- MASSF: the actual flow rate

EQ (General)

Selection of equilibrium equation (type: Input).

0	None	No equilibrium equation
1	Yes	White equilibrium equation is activated.

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if $MODEDP = 0$.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

DTEQ (Parameters)

Temperature deviation from equilibrium (type: Measurable).

The DTEQ parameter allows you to account for a deviation from equilibrium by calculating the equilibrium constant at a temperature different ($T + \text{DTEQ}$) from the outlet stream temperature T .

Attention: When $\text{DTEQ} > 100$ C, no equilibrium constraint is generated. This has the same effect as setting the parameter EQi to 0.

EXAMPLE : NH₃ SYNTHESIS REACTOR

Attention: see model `bwhval.bls` in the directory "`\examples\bwhval`".

The same basic example is taken for the following unit types: **BWHVAL**, **RCTVAL**, **RKPVAL**

This example deals with a chemical reactor for the synthesis of NH₃.

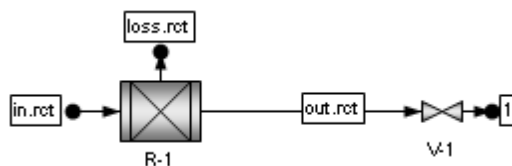


FIGURE 8 - BWHVAL EXAMPLE: SYNTHESIS OF NH₃

There are 2 UNITS :

- R-1:
 - type=BWHVAL
 - MODE='Atomic and Energy Balance'
 - MODEDP='DP variable'
 - White Equilibrium=Yes
 - REFDP=IN.RCT
 - DP=1.1 bar
 - DTEQ=25 K
- V-1:
 - type=DPVAL
 - MODE='Mass and Energy Balance'
 - MODEDP='DP variable'
 - K=-1

4 Streams must be defined:

STREAMS THERMOD		CONNECTIONS
IN.RCT	RCMOD	To R-1
OUT.RCT	RCMOD	From R-1 To V-1
LOSS.RCT	THERMAL STREAM	From R-1
1	RCMOD	From V-1

The Thermod RCMOD uses the default methods and the following compounds: NH₃, N₂, H₂, AR.

Table below gives the list of measurements.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
1_P	STREAM	1	P	100.00	5.00	bar
IN.RCT_MFAR	STREAM	IN.RCT	MFAR	5.7000	-5.000	%
IN.RCT_MFH2	STREAM	IN.RCT	MFH2	70.000	-5.000	%
IN.RCT_MFN2	STREAM	IN.RCT	MFN2	21.000	-5.000	%
IN.RCT_MFNH3	STREAM	IN.RCT	MFNH3	3.3000	-5.000	%
IN.RCT_MOLF	STREAM	IN.RCT	MOLF	0.78000E+06	-10.00	Nm3/h
IN.RCT_P	STREAM	IN.RCT	P	200.00	1.00	bar
IN.RCT_T	STREAM	IN.RCT	T	282.00	5.00	C
OUT.RCT_MFAR	STREAM	OUT.RCT	MFAR	6.0000	-5.000	%
OUT.RCT_MFH2	STREAM	OUT.RCT	MFH2	61.000	-5.000	%
OUT.RCT_MFN2	STREAM	OUT.RCT	MFN2	17.000	-5.000	%
OUT.RCT_MFNH3	STREAM	OUT.RCT	MFNH3	16.000	-5.000	%
OUT.RCT_P	STREAM	OUT.RCT	P	198.00	1.00	bar
OUT.RCT_T	STREAM	OUT.RCT	T	470.00	CST	C
R-1_DP	UNIT	R-1	DP	1.1000	0.500E-01	bar
R-1_DTEQ	UNIT	R-1	DTEQ	10.000	20.0	K
V-1_DP	UNIT	V-1	DP	95.000	5.00	bar

TABLE 5 - MEASUREMENTS FOR BWHVAL EXAMPLE

The results are (file r6v):

Number of tags	17
Number of equations	19
Number of unmeasured variables	12
Number of measured variables	16
Number of variables considered constant	1
Obvious number of redundancies	7
Total number of redundancies	7
Number of trivial redundancies	0

Validation and reconciliation results

Objective function	2.14360
Chi-square test value	14.0700
Sum of squares of residues	0.263445E-008
Number of active bounds	0
Number of variables close to bounds	0
Number of eliminated tags	0
Number of flagged tags	0
Number of filtered tags	0

TAGS IN THE MODEL

Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
1_P	100.00	5.00	101.70	3.55	0.12	bar
IN.RCT_MFAR	5.7000	5.00 %	5.5097	3.55 %	0.45	%
IN.RCT_MFH2	70.000	5.00 %	70.436	0.901 %	0.02	%
IN.RCT_MFN2	21.000	5.00 %	20.752	2.88 %	0.06	%
IN.RCT_MFNH3	3.3000	5.00 %	3.3024	4.99 %	0.00	%
IN.RCT_MOLF	0.78000E+06	10.0 %	0.78002E+06	10.0 %	0.00	Nm3/h
IN.RCT_P	200.00	1.00	199.51	0.704	0.24	bar
IN.RCT_T	282.00	5.00	282.00	5.00	0.00	C
OUT.RCT_MFAR	6.0000	5.00 %	6.1934	3.53 %	0.42	%
OUT.RCT_MFH2	61.000	5.00 %	60.563	1.56 %	0.02	%
OUT.RCT_MFN2	17.000	5.00 %	17.123	3.92 %	0.02	%
OUT.RCT_MFNH3	16.000	5.00 %	16.121	4.54 %	0.02	%
OUT.RCT_P	198.00	1.00	198.41	0.704	0.17	bar
OUT.RCT_T	470.00	CST	470.00			C
R-1_DP	1.1000	0.500E-01	1.1012	0.500E-01	0.00	bar
R-1_DTEQ	10.000	20.0	24.229	5.94	0.51	K
V-1_DP	95.000	5.00	96.704	3.55	0.12	bar

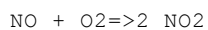
TABLE 6 - RESULTS OF BWHVAL EXAMPLE

SECTION 7: CND-HNO3: NITROUS GAS CONDENSER

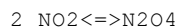
PURPOSE

This UNIT models the cooling and partial condensation of a nitrous gas mixture, taking into account reactions in the gas phase:

peroxidation:

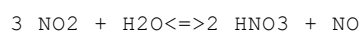


dimerization:



as well as at the liquid-vapor interface :

absorption:



The extents of these reactions are calculated on basis of BODENSTEIN kinetic equations.

The condensed liquid is a mixture of nitric acid and water and is assumed to be in L/V equilibrium with the remaining gas.

SYMBOLS

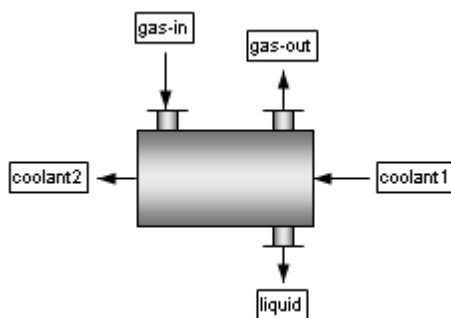


FIGURE 9 - SYMBOLS FOR UNIT CND-HNO3

CONNECTION RULES

INLET STREAMS (2)

- shell side, nitrous gas mixture (state V, LV, LLV or WLV)
- tube side, cooling media (any state)

OUTLET STREAMS (3)

- shell side, remaining nitrous gas mixture (state V)
- shell side, nitric acid aqueous mixture (state L)
- tube side, cooling media (any state)

THERMOD:

Both nitrous gas material streams must use the same THERMOD, which must contain the following compounds in any order : NO, NO₂, N₂O₄, O₂, N₂, H₂O, HNO₃. Any other compound is considered as inert.

The THERMOD of the nitric acid mixture must contain the following compounds in this order: H₂O, HNO₃.

Both coolant streams(inlet and outlet) must use the same THERMOD.

REACTIONS:

The reactions are part of the model and must not be defined in the Reactions page.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
LOAD	Parameters	Result	Heat load of the exchanger
VOL	Parameters	Measurable	Shell volume
HA	Parameters	Measurable	Transfer coefficient
DTSIDE1,2	Parameters	Measurable	Temperature difference on each side
ACTi	Parameters	Measurable	Activity coefficient (Water, Nitric acid)
MODEDPi	Parameters	Input	Pressure equation
DPi	Parameters	Measurable	Pressure drop
EXTENTi	Parameters	Measurable	Extent of reactions and condensations

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	Atomic and Energy Balance	Atomic and energy balances are taken into account
1	Atomic Balance only	Only atomic balances are taken into account

LOAD (Parameters)

Heat load of the exchanger (type: Result).

This parameter is the amount of heat transferred by the unit.

VOL (Parameters)

Shell volume (type: Measurable)

The shell volume is taken into account into the Bodenstein kinetics.

HA (Parameters)

Transfer coefficient (type: Measurable)

The heat transfer coefficient is defined as follows:

$$LOAD = HA \cdot LNTD$$

$$LNTD = \frac{DTSIDE1 - DTSIDE2}{\ln(DTSIDE1/DTSIDE2)}$$

where:

- DTSIDE1 and DTSIDE2 are the temperature differences on each side of the heat exchanger

DTSIDE1,2 (Parameters)

Temperature difference on each side (type: Measurable)

These parameters define the temperature difference of the streams at each side of the exchanger:

$$DTSIDE1 = Ti1 - To2$$

$$DTSIDE2 = To1 - Ti2$$

ACTi (Parameters)

Activity coefficient (Water, Nitric acid) (type: Measurable)

The activity coefficients are defined as:

$$ACT_i = \frac{PP_i}{x_i \cdot P_i^*}$$

where:

- P_i : the partial pressure of compound i
- x_i : the molar fraction of compound i in the liquid phase
- P_i^* : the saturation pressure of compound i
- i: 1=HNO₃, 2=H₂O

These activity coefficients can be determined according to the following formulas :

$$ACT_1 = \frac{(A_1 T^2 + B_1 T + C_1) x_2 + (D_1 T^3 + E_1 T^2 + F_1 T + G_1)}{x_1 P_1^*}$$

$$ACT_2 = \frac{A_2 \exp(B_2 T) \exp((C_2 T + D_2 + E_2 T^2) x_2)}{x_2 P_2^*}$$

where:

- the temperature T is expressed in °C
- the pressure is expressed in mmHg
- the coefficients $A_i \dots G_i$ have the following values:

	i=1 (H ₂ O)	i=2 (HNO ₃)
A_i	-0.082377958	0.000095684
B_i	1.271367016	0.087355345
C_i	-9.039063345	-0.002051086
D_i	0.000563725	12.648391590
E_i	0.016013187	-0.000441081
F_i	-0.058557541	
G_i	8.991622939	

Actual activity coefficients might differ from these predicted values due to the limited performance of the condensor. It is therefore recommended to use the difference between the actual and predicted values as a KPI (Key Performance Indicator) of the condenser efficiency.

EXTENT (Parameters)

Extent of reactions and condensations (type: Measurable).

The Extents are defined (on molar basis) as follows:

- Extent1: amount of O₂ consumed in peroxidation
- Extent2: amount of N₂O₄ produced by dimerization
- Extent3: amount of HNO₃ produced by absorption
- Extent4: amount of condensed H₂O
- Extent5: amount of condensed HNO₃

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

A value of MODEDP_i has to be given for both outlet streams.

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{in} - P_{out}$$

This equation is not generated if MODEDP = 0.

where:

- P_{in} : the pressure of the inlet stream
- P_{out} : the pressure of the outlet stream

EXAMPLE : NITRIC ACID PROCESS

Attention: see model nitric.bls in the directory "\examples\nitric.

SECTION 8: COLVAL: DISTILLATION COLUMN

PURPOSE

This type of UNIT allows you to consider mass balances, heat balance and saturation equations for the outlet streams.

It is especially useful to model distillation columns or any other operation unit where some outlet streams are saturated liquid or vapor.

SYMBOLS

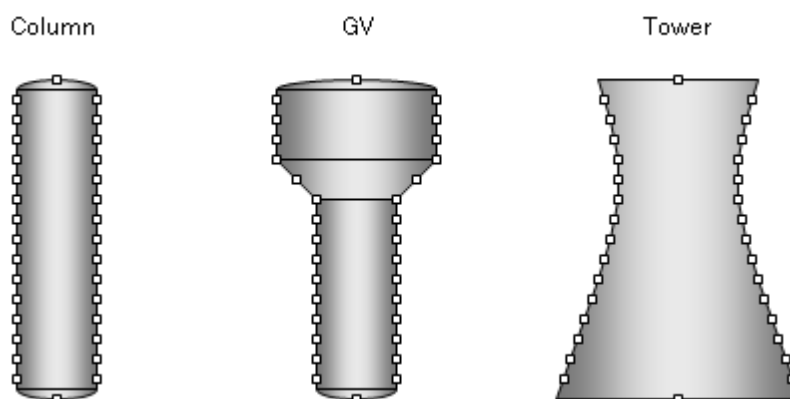


FIGURE 10 - SYMBOLS FOR UNIT COLVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 10
- Type: material (at least one), thermal, mechanical

OUTLET STREAMS

- Maximum number: 10
- Type: material (at least one), thermal, mechanical

The outlet STREAMS can be in L, V or LV state. You must use the L state for each outlet stream that is specified at the bubble point, and the V state for each outlet stream that is specified at the dew point.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
COMPMODE	General	Input	Compound balance mode
REFDP	Parameters	Input	Reference stream for pressure drop
VFRACi	Parameters	Input	Saturation state
MODEDPi	Parameters	Input	Pressure equation
DPi	Parameters	Measurable	Pressure drop

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	All equations	Mass, energy balances and saturation constraints are taken into account
1	Mass Balance only	Only mass balances are taken into account
2	Mass and Energy balance	Mass and energy balances are taken into account
3	Mass Balance and Saturation	Mass balances and saturation constraints are taken into account

COMPMODE (General)

Compound balance mode (type: Input).

1	Enabled	Compound balances are taken into account
3	None	Only global mass balance (without compound balances) is taken into account.

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

VFRACi (Parameters)

Saturation state (type: Input).

A value of VFRAC must be given for each outlet Stream, although it will only be meaningful for material STREAMS:

-1	-	No saturation constraint for the corresponding outlet stream
0	-	The corresponding stream is at bubble point. It must be defined as Liquid
1	-	The corresponding stream is at dew point. It must be defined as Vapor

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

EXAMPLE : TWO INTERCONNECTED COLUMNS

Attention: see model colval.bls in the directory "examples\colval". A unit of type COLVAL is also used to model a de-aerator in the example coal.bls (coal fired power plant).

In this example, a mixture of propane, pentane and heptane is distilled into three streams of high purity.

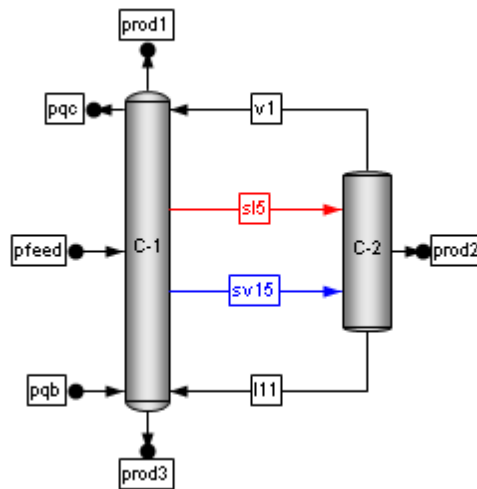


FIGURE 11 - COLVAL EXAMPLE: TWO INTERCONNECTED COLUMNS

There are 2 UNITS:

- C-1
 - type = COLVAL
 - MODE='All equations'
 - VFRAC1=0 (PROD1 is a saturated liquid)
 - VFRAC2=0 (PROD3 is a saturated liquid)
 - VFRAC3=0 (SL5 is a saturated liquid)
 - VFRAC4=1 (SV15 is a saturated vapor)
- C-2
 - type = COLVAL
 - MODE='All equations'
 - VFRAC1=0 (PROD2 is a saturated liquid)
 - VFRAC2=0 (L11 is a saturated liquid)
 - VFRAC3=1 (V1 is a saturated liquid)

10 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
PFEED	VENTMOD	to COLUMN
PROD1	VENTMOD	from COLUMN
PROD3	VENTMOD	from COLUMN
SL5	VENTMOD	from COLUMN
SV15	VENTMOD	from COLUMN to COLUMN1
PROD2	VENTMOD	from COLUMN1
L11	VENTMOD	from COLUMN1 to COLUMN
V1	VENTMOD	from COLUMN1 to COLUMN
PQB	THERMAL STREAM	to COLUMN
PQC	THERMAL STREAM	from COLUMN

The thermodynamic calculation methods are the ones used by default.

The compounds associated to the THERMOD VENTMOD are: C3, N-C5, N-C7.

Table below gives the list of measurements.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
L11_MASSF	STREAM	L11	MASSF	0.25000	-10.00	kg/s
L11_MFC3	STREAM	L11	MFC3	0.50000E-01	-10.00	%
L11_MFN-C5	STREAM	L11	MFN-C5	30.000	-1000.	%
L11_MFN-C7	STREAM	L11	MFN-C7	70.000	-1000.	%
L11_P	STREAM	L11	P	9.8000	0.200	bar
L11_T	STREAM	L11	T	143.30	2.00	C
PFEED_MASSF	STREAM	PFEED	MASSF	0.72000	-10.00	kg/s
PFEED_MFC3	STREAM	PFEED	MFC3	30.000	-2.000	%
PFEED_MFN-C5	STREAM	PFEED	MFN-C5	40.000	-2.000	%
PFEED_MFN-C7	STREAM	PFEED	MFN-C7	30.000	-2.000	%
PFEED_P	STREAM	PFEED	P	10.000	0.200	bar
PFEED_T	STREAM	PFEED	T	83.300	2.00	C
PQB_LOAD	STREAM	PQB	LOAD	440.00	-5.000	kW
PQC_LOAD	STREAM	PQC	LOAD	385.00	-5.000	kW
PROD1_MASSF	STREAM	PROD1	MASSF	0.13000	-10.00	kg/s
PROD1_MFC3	STREAM	PROD1	MFC3	99.920	0.100	%
PROD1_MFN-C5	STREAM	PROD1	MFN-C5	0.80000E-01	0.100E-01	%
PROD1_MFN-C7	STREAM	PROD1	MFN-C7	0.0000	CST	%
PROD1_P	STREAM	PROD1	P	9.5000	0.200	bar
PROD1_T	STREAM	PROD1	T	24.800	2.00	C
PROD2_MASSF	STREAM	PROD2	MASSF	0.29000	-10.00	kg/s
PROD2_MFC3	STREAM	PROD2	MFC3	0.50000E-03	-10.00	%
PROD2_MFN-C5	STREAM	PROD2	MFN-C5	98.330	0.100	%
PROD2_MFN-C7	STREAM	PROD2	MFN-C7	1.6700	0.100	%
PROD2_P	STREAM	PROD2	P	9.7500	0.200	bar
PROD2_T	STREAM	PROD2	T	124.30	2.00	C
PROD3_MASSF	STREAM	PROD3	MASSF	0.30000	-10.00	kg/s
PROD3_MFC3	STREAM	PROD3	MFC3	0.40000E-01	-10.00	%
PROD3_MFN-C5	STREAM	PROD3	MFN-C5	2.1000	0.100	%
PROD3_MFN-C7	STREAM	PROD3	MFN-C7	97.860	0.100	%
PROD3_P	STREAM	PROD3	P	10.000	0.200	bar
PROD3_T	STREAM	PROD3	T	198.60	2.00	C
SL5_MASSF	STREAM	SL5	MASSF	0.49000	-10.00	kg/s
SL5_MFC3	STREAM	SL5	MFC3	70.000	-1000.	%
SL5_MFN-C5	STREAM	SL5	MFN-C5	30.000	-1000.	%
SL5_MFN-C7	STREAM	SL5	MFN-C7	0.40000E-02	-10.00	%
SL5_P	STREAM	SL5	P	9.7000	0.200	bar
SL5_T	STREAM	SL5	T	34.200	2.00	C
SV15_MASSF	STREAM	SV15	MASSF	0.60000	-10.00	kg/s
SV15_MFC3	STREAM	SV15	MFC3	1.0000	-10.00	%
SV15_MFN-C5	STREAM	SV15	MFN-C5	30.000	-1000.	%
SV15_MFN-C7	STREAM	SV15	MFN-C7	70.000	-1000.	%
SV15_P	STREAM	SV15	P	9.9000	0.200	bar
SV15_T	STREAM	SV15	T	146.70	2.00	C
V1_MASSF	STREAM	V1	MASSF	0.55000	-10.00	kg/s
V1_MFC3	STREAM	V1	MFC3	70.000	-1000.	%
V1_MFN-C5	STREAM	V1	MFN-C5	30.000	-1000.	%
V1_MFN-C7	STREAM	V1	MFN-C7	0.50000E-03	-10.00	%
V1_P	STREAM	V1	P	9.7000	0.200	bar
V1_T	STREAM	V1	T	34.200	2.00	C

TABLE 7 - MEASUREMENTS FOR COLVAL EXAMPLE

The results are (file r6v):

Number of tags	50
Number of equations	54
Number of unmeasured variables	31
Number of measured variables	49
Number of variables considered constant	2
Obvious number of redundancies	23
Total number of redundancies	23
Number of trivial redundancies	0

Validation and reconciliation results

Objective function	13.3834
Chi-square test value	35.1800
Sum of squares of residues	0.628052E-006
Number of active bounds	0
Number of variables close to bounds	0
Number of eliminated tags	0
Number of flagged tags	1
Number of filtered tags	0

TAGS IN THE MODEL

Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
L11_MASSF	0.25000	10.0 %	0.25150	8.07 %	0.00	kg/s
L11_MFC3	0.50000E-01	10.0 %	0.49965E-01	10.0 %	0.00	%
L11_MFN-C5	30.000	0.100E+04%	63.557	5.24 %	0.01	%
L11_MFN-C7	70.000	0.100E+04%	36.393	9.15 %	0.00	%
L11_P	9.8000	0.200	9.8814	0.193	0.17	bar
L11_T	143.30	2.00	141.82	1.76	0.55	C
PFEED_MASSF	0.72000	10.0 %	0.73727	4.67 %	0.06	kg/s
PFEED_MFC3	30.000	2.00 %	29.938	1.68 %	0.01	%
PFEED_MFN-C5	40.000	2.00 %	40.101	1.41 %	0.02	%
PFEED_MFN-C7	30.000	2.00 %	29.960	1.67 %	0.00	%
PFEED_P	10.000	0.200	10.000	0.200	0.00	bar
PFEED_T	83.300	2.00	83.294	1.95	0.00	C
PQB_LOAD	440.00	5.00 %	439.80	3.44 %	0.00	kW
PQC_LOAD	385.00	5.00 %	385.15	3.86 %	0.00	kW
PROD1_MASSF	0.13000	10.0 %	0.13501	4.98 %	0.15	kg/s
PROD1_MFC3	99.920	0.100	99.920	0.995E-02	0.00	%
PROD1_MFN-C5	0.80000E-01	0.100E-01	0.79986E-01	0.995E-02	0.00	%
PROD1_MFN-C7	0.0000	CST	0.0000			%
PROD1_P	9.5000	0.200	9.4922	0.185	0.00	bar
PROD1_T	24.800	2.00	24.989	0.762	0.01	C
PROD2_MASSF	0.29000	10.0 %	0.29768	4.88 %	0.07	kg/s
PROD2_MFC3	0.50000E-03	10.0 %	0.50000E-03	10.0 %	0.00	%
PROD2_MFN-C5	98.330	0.100	98.331	0.707E-01	0.00	%
PROD2_MFN-C7	1.6700	0.100	1.6687	0.707E-01	0.00	%
PROD2_P	9.7500	0.200	9.7431	0.178	0.00	bar
PROD2_T	124.30	2.00	124.47	0.919	0.01	C
PROD3_MASSF	0.30000	10.0 %	0.30458	4.95 %	0.02	kg/s
PROD3_MFC3	0.40000E-01	10.0 %	0.39993E-01	10.0 %	0.00	%
PROD3_MFN-C5	2.1000	0.100	2.0991	0.707E-01	0.00	%
PROD3_MFN-C7	97.860	0.100	97.861	0.707E-01	0.00	%
PROD3_P	10.000	0.200	10.018	0.172	0.01	bar
PROD3_T	198.60	2.00	198.29	1.02	0.02	C
SL5_MASSF	0.49000	10.0 %	0.53482	4.04 %	0.84	kg/s
SL5_MFC3	70.000	0.100E+04%	93.063	3.59 %	0.00	%
SL5_MFN-C5	30.000	0.100E+04%	6.9330	48.2 %	0.01	%
SL5_MFN-C7	0.40000E-02	10.0 %	0.40001E-02	10.0 %	0.00	%
SL5_P	9.7000	0.200	9.8970	0.190	0.97	bar
SL5_T	* 34.200	2.00	29.259	1.38	6.10	C
SV15_MASSF	0.60000	10.0 %	0.51139	5.42 %	2.18	kg/s
SV15_MFC3	1.0000	10.0 %	1.0034	9.97 %	0.00	%
SV15_MFN-C5	30.000	0.100E+04%	81.273	1.94 %	0.03	%
SV15_MFN-C7	70.000	0.100E+04%	17.723	8.90 %	0.01	%
SV15_P	9.9000	0.200	9.8012	0.190	0.24	bar
SV15_T	146.70	2.00	148.55	1.61	0.85	C
V1_MASSF	0.55000	10.0 %	0.49703	6.92 %	0.93	kg/s
V1_MFC3	70.000	0.100E+04%	97.821	0.626 %	0.00	%
V1_MFN-C5	30.000	0.100E+04%	2.1783	28.1 %	0.01	%
V1_MFN-C7	0.50000E-03	10.0 %	0.49999E-03	10.0 %	0.00	%
V1_P	9.7000	0.200	9.6786	0.200	0.01	bar
V1 T	34.200	2.00	34.795	1.99	0.09	C

TABLE 8 - RESULTS OF COLVAL EXAMPLE

SECTION 9: COMVAL: GAS COMPRESSOR

PURPOSE

This type of UNIT allows to take the mass and energy balances of a gas compressor into account.

Information on the isentropic (or polytropic) efficiency and the mechanical power can be given as measurements.

SYMBOLS

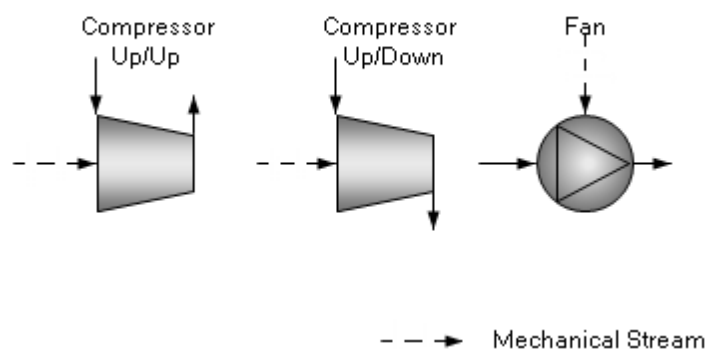


FIGURE 12 - SYMBOLS FOR UNIT COMVAL

CONNECTION RULES

INLET STREAMS

- 1 material
- 1 mechanical

OUTLET STREAMS

- 1 material

THERMOD

Both material streams must use the same THERMOD.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation

EFFIC	Parameters	Measurable	Isentropic efficiency
POLYEFFI	Parameters	Measurable	Polytropic efficiency
DP	Parameters	Measurable	Pressure increase
PRATIO	Parameters	Measurable	Pressure ratio

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	All equations	Mass and energy balances are taken into account. The isentropic efficiency equation is generated as well
1	Mass Balance only	Only mass balances are taken into account
2	Mass and Energy Balance	Mass and energy balances are taken into account

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

EFFIC (Parameters)

Isentropic efficiency (type: Measurable).

This parameter is only relevant in MODE 0.

POLYEFFI (Parameters)

Polytropic efficiency (type: Measurable).

When negative, this parameter is ignored. When positive, an additional equation is generated to calculate the polytropic efficiency.

Attention: the formulation of the polytropic efficiency assumes that the mixture behaves like an ideal gas.

DP (Parameters)

Pressure increase (type: Measureable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{out} - P_{in}$$

This equation is not generated if MODEDP = 0.

PRATIO (Parameters)

Pressure ratio (type: Measurable)

$$PRATIO = P_{out}/P_{in}$$

This equation is not generated if MODEDP = 0.

EXAMPLE : AIR COMPRESSOR

Attention: see model comval.bls in the directory "examples\comval".

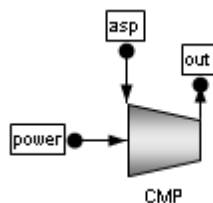


FIGURE 13 - COMVAL EXAMPLE: AIR COMPRESSOR

There is 1 UNIT:

- CMP:
 - type=COMVAL
 - MODE='All equations'
 - MODEDP='DP Variable'
 - POLYEFFI = 80 %

3 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
ASP	AIR	to CMP
POWER	MECHANICAL STREAM	to CMP
OUT	AIR	from CMP

The THERMOD AIR uses the default thermodynamic methods and two compounds: N2, O2

The list of measurements is given in table below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
ASP_MFN2	STREAM	ASP	MFN2	79.000	CST	%
ASP_MFO2	STREAM	ASP	MFO2	21.000	CST	%
ASP_MOLF	STREAM	ASP	MOLF	19500.	-5.000	Nm3/h
ASP_P	STREAM	ASP	P	1.0130	-2.000	bar
ASP_T	STREAM	ASP	T	25.000	1.00	C
CMP_EFFIC	UNIT	CMP	EFFIC	80.000	10.0	%
CMP_POLYEFFI	UNIT	CMP	POLYEFFI	80.000	10.0	%
OUT_P	STREAM	OUT	P	3.0000	CST	bar
OUT_T	STREAM	OUT	T	150.00	1.00	C
POWER_POWER	STREAM	POWER	POWER	950.00	-10.00	kW

TABLE 9 - MEASUREMENTS FOR COMVAL EXAMPLE

The results are (file r6v):

Number of tags	10					
Number of equations	9					
Number of unmeasured variables	6					
Number of measured variables	7					
Number of variables considered constant	3					
Obvious number of redundancies	3					
Total number of redundancies	3					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	1.12498					
Chi-square test value	7.81000					
Sum of squares of residues	0.866434E-013					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
ASP_MFN2	79.000	CST	79.000			%
ASP_MFO2	21.000	CST	21.000			%
ASP_MOLF	19500.	5.00 %	19730.	4.48 %	0.06	Nm3/h
ASP_P	1.0130	2.00 %	1.0171	1.94 %	0.04	bar
ASP_T	25.000	1.00	24.857	0.991	0.02	C
CMP_EFFIC	80.000	10.0	85.025	2.07	0.25	%
CMP_POLYEFFI	80.000	10.0	87.001	1.81	0.49	%
OUT_P	3.0000	CST	3.0000			bar
OUT_T	150.00	1.00	150.11	0.994	0.01	C
POWER_POWER	950.00	10.0 %	902.29	4.57 %	0.25	kW

TABLE 10 - RESULTS OF COMVAL EXAMPLE

SECTION 10: CRUOPT: OPTIONS FOR CRUDE FRACTIONS IDENTIFICATION

PURPOSE

The **CRUOPT** unit is a mathematical unit used to specify various parameters of the procedure used to identify petroleum fractions properties and to link distillation curves and stream compositions.

If no **CRUOPT** unit is defined, the default settings are used.

SYMBOLS



FIGURE 14 - SYMBOLS FOR UNIT CRUOPT

CONNECTION RULES

No connections are allowed on a **CRUOPT**.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODECURV	General	Input	D86/TBP continuity curve mode
MODEELIM	General	Input	Auto elimination of TBP/ASTM points
MODEIPFP	General	Input	IBP/FBP extrapolation mode
SMOOTH	Parameters	Input	Precision for Smooth equation
RKUOP	Parameters	Input	Precision for KUOP equation
PLFLOW	Parameters	Input	Precision for LF Low
PLFHIGH	Parameters	Input	Precision for LF High
PLFTRESH	Parameters	Input	LF Threshold

UNIT PARAMETERS

MODECURV (General)

D86/TBP continuity curve mode (type: Input).

The MODECURV parameter selects the interpolation method for matching the stream composition to the distillation curve.

0	Linear Smoothing	The Linear Smoothing identifies the composition by adding continuity factors between successive compound fractions so as to insure a smooth evolution of the distillation curve. This method tends to slightly flatten the curve and is kept for compatibility reason
1	Compositional Interpolation (default)	The Compositional Interpolation identifies the composition by adding compound fractions estimated on basis of the distillation curve

MODEELIM (General)

Auto elimination of TBP/ASTM points (type: Input)

0	No automatic elimination (default)	the automatic elimination will not remove the distillation points that exhibits a decrease of the temperature but will stop the calculation
1	Automatic elimination (ValiFilter license required)	the automatic elimination will remove the distillation points that exhibits a decrease of the temperature

MODEIPFP (General)

IBP/FBP extrapolation mode (type: Input).

0	Linear extrapolation	The following rules are applied to extrapolate the Initial and Final boiling points from the TBP curve: 1. Initial Boiling Point (IBP): ■ If the lowest measured point from the TBP curve is less than 10%, VALI uses a linear extrapolation from the 2 first measured points: $IBP = T_1 + \frac{T_2 - T_1}{V_2 - V_1} V_1$ ■ Otherwise, VALI assumes that IBP = 273.15 K 2. Final Boiling Point (FBP): ■ If the largest measured point from the TBP curve is more than 90%, VALI uses a linear extrapolation from the 2 last measured points: $FBP = T_n + \frac{T_n - T_{n-1}}{V_n - V_{n-1}} (100 - V_n)$ ■ Otherwise, VALI assumes that FBP = 2273.15 K
1	Gaussian Approximation	Approximation of the TBP distribution with a Cumulative Gaussian distribution function is a good alternative to extrapolate incomplete TBP curves. This approach allows here to predict the Initial and Final boiling points.

SMOOTH (Parameters)

Precision for Smooth equation (type: Input).

Default value = 0.30 (30 %)

The SMOOTH parameter is only used in Linear Smoothing mode (MODECURV = 0). It represents the precision associated to each smoothing factor.

RKUOP (Parameters)

Precision for KUOP equation (type: Input).

Default value = 0.01 (1 %) - In releases until 2004, the default precision was 5%.

When identifying pseudo-compounds, continuity factors are automatically introduced on the KUOP (also known as Watson factor) of successive compounds. These factors are calculated as the ratio of the KUOPs and measured as 1 +/- RKUOP %.

PLFLOW (Parameters)

Precision for LF Low (type: Input).

PLFHIGH (Parameters)

Precision for LF High (type: Input).

PLFTRESH (Parameters)

LF Threshold (type: Input).

These three parameters control the precision used for the Composition Interpolation (MODECURV = 1).

The precision of compound fraction is set to PLFLOW for small amounts (fraction below PLFTRESH) and to PLFHIGH for higher amounts.

The default values are:

- PLFLOW = 0.10 (10%)
- PLFHIGH = 0.30 (30 %)
- PLFTRESH = 0.01 (1 %)

SECTION 11: CRUVAL: CRUDE FRACTIONS IDENTIFICATION

PURPOSE

The **CRUVAL** unit is devoted to the identification of petroleum fractions (or pseudo-compounds) to be used for the adequate modeling of petroleum cuts. Measurements of density and distillation curves (TBP or D86) are usually defined on the streams connected to the unit.

The **CRUVAL** is basically a black box like the **BBXVAL**, but when used in Mass balance mode, it also allows identifying the properties of the pseudo-compounds used in the streams connected to the unit.

A typical way of using the **CRUVAL** is to perform two steps:

- mass balance and identification of the pseudo-compounds
- complete heat and mass balance using the pseudo-compounds identified in the first step

The way distillation curves and compounds/composition identification are handled can be adjusted through the mathematical unit **CRUOPT**.

SYMBOLS

Column

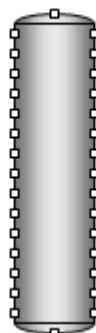


FIGURE 15 - SYMBOLS FOR UNIT CRUVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 10
- Type: material, thermal

OUTLET STREAMS

- Maximum number: 10
- Type: material, thermal

Attention: At least one inlet and one outlet material streams must be defined.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
COMPMODE	General	Input	Compound balance mode
IDENT	General	Input	Identification mode
REFDP	Parameters	Input	Reference stream for pressure drop
MODEDPi	Parameters	Input	Pressure equation
DPi	Parameters	Measurable	Pressure drop
DPXi	Parameters	Input	Nominal pressure drop
MASSFxi	Parameters	Input	Nominal flow rate
MODEDPi	Parameters	Input	Pressure equations

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Mass and Energy Balance	Mass and energy balances are taken into account
1	Mass Balance only	Only mass balances are taken into account

COMPMODE (General)

Compound balance mode (type: Input).

1	Enabled	Compound balances are taken into account
3	None	Only global mass balance (without compound balances) is taken into account.

IDENT (General)

Identification mode (type: Input)

0	No Identification	No identification is performed
1	Identification	Identification is performed
2	Identification with parabolic KUOP curve	Identification is performed using KUOP parabolic curve

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

MODEDP (Parameters)

Pressure equation (type: Input)

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * (MASSF / MASSFX)^2$$

where:

- CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%)
- DPX: the nominal pressure drop
- MASSFX: the nominal flow rate
- MASSF: the actual flow rate

A value of MODEDP_i has to be given for each outlet stream, although it is only meaningful for material streams.

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when MODEDP = 2.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when MODEDP = 2.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when MODEDP = 2.

EXAMPLE : CRUDE DISTILLATION COLUMN

Note: see model `cruval.bls` in the directory "`examples\cruval`".

In this example, the **CRUVAL** type of unit is used to model an atmospheric crude distillation column. Only the mass balances are taken into account.

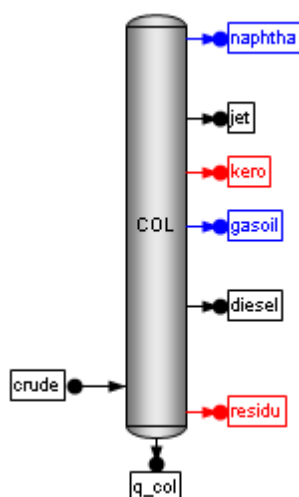


FIGURE 16 - CRUVAL EXAMPLE: CDU

There is 1 UNIT:

- COL:
 - type=CRUVAL
 - MODE='Mass Balance only'
 - MODEDPi='None'

7 STREAMS must be defined:

STREAMS	THERMOD	CONNECTIONS
CRUDE	CRUDE	to COL
NAPHTHA	CRUDE	from COL
JET	CRUDE	from COL
KERO	CRUDE	from COL
GASOIL	CRUDE	from COL
DIESEL	CRUDE	from COL
RESIDU	CRUDE	from COL

The thermodynamic calculation methods are the ones used by default, as we are only concerned with the mass balances of the unit.

The compounds used in the Thermod CRUDE are:

C3, I-C4, N-C4, 22MC3, N-C5, FR050, FR060, FR070, FR080, FR090, FR100, FR120, FR140, FR160, FR180, FR200, FR220, FR240, FR260, FR280, FR300, FR330, FR360, FR390, FR420, FR450, FR500, FR600, FR700

The list of measurements is given in table below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
CRUDE_D60	STREAM	CRUDE	COLDDENS	0.93180	0.500E-03	D60/60
CRUDE_MASSF	STREAM	CRUDE	MASSF	567.00	-5.000	t/h
CRUDE_P	STREAM	CRUDE	P	4.0000	-5.000	bar
CRUDE_T	STREAM	CRUDE	T	703.40	3.60	F
DIESEL_D60	STREAM	DIESEL	COLDDENS	0.95000	0.500E-03	D60/60
DIESEL_D86V0	STREAM	DIESEL	D86V000	501.80	5.40	F
DIESEL_D86V10	STREAM	DIESEL	D86V010	604.40	5.40	F
DIESEL_D86V100	STREAM	DIESEL	D86V100	770.00	5.40	F
DIESEL_D86V30	STREAM	DIESEL	D86V030	636.80	5.40	F
DIESEL_D86V5	STREAM	DIESEL	D86V005	584.60	5.40	F
DIESEL_D86V50	STREAM	DIESEL	D86V050	671.00	5.40	F
DIESEL_D86V70	STREAM	DIESEL	D86V070	698.00	5.40	F
DIESEL_D86V90	STREAM	DIESEL	D86V090	739.40	5.40	F
DIESEL_D86V95	STREAM	DIESEL	D86V095	761.00	5.40	F
DIESEL_MASSF	STREAM	DIESEL	MASSF	51.100	-5.000	t/h
DIESEL_P	STREAM	DIESEL	P	3.9400	-5.000	bar
DIESEL_T	STREAM	DIESEL	T	635.00	3.60	F
GASOIL_D60	STREAM	GASOIL	COLDDENS	0.90820	0.500E-03	D60/60
GASOIL_D86V0	STREAM	GASOIL	D86V000	338.00	5.40	F
GASOIL_D86V10	STREAM	GASOIL	D86V010	474.80	5.40	F
GASOIL_D86V100	STREAM	GASOIL	D86V100	626.00	5.40	F
GASOIL_D86V30	STREAM	GASOIL	D86V030	514.40	5.40	F
GASOIL_D86V5	STREAM	GASOIL	D86V005	458.60	5.40	F
GASOIL_D86V50	STREAM	GASOIL	D86V050	532.40	5.40	F
GASOIL_D86V70	STREAM	GASOIL	D86V070	555.80	5.40	F
GASOIL_D86V90	STREAM	GASOIL	D86V090	593.60	5.40	F
GASOIL_D86V95	STREAM	GASOIL	D86V095	608.00	5.40	F
GASOIL_MASSF	STREAM	GASOIL	MASSF	69.400	-5.000	t/h
GASOIL_P	STREAM	GASOIL	P	3.8400	-5.000	bar
GASOIL_T	STREAM	GASOIL	T	552.20	3.60	F
JET_D60	STREAM	JET	COLDDENS	0.85710	0.500E-03	D60/60
JET_D86V0	STREAM	JET	D86V000	291.20	5.40	F
JET_D86V10	STREAM	JET	D86V010	336.20	5.40	F
JET_D86V30	STREAM	JET	D86V030	354.20	5.40	F
JET_D86V5	STREAM	JET	D86V005	327.20	5.40	F
JET_D86V50	STREAM	JET	D86V050	374.00	5.40	F
JET_D86V70	STREAM	JET	D86V070	392.00	5.40	F
JET_D86V90	STREAM	JET	D86V090	435.20	5.40	F
JET_D86V95	STREAM	JET	D86V095	458.60	5.40	F
JET_MASSF	STREAM	JET	MASSF	73.100	-5.000	t/h
JET_P	STREAM	JET	P	3.6500	-5.000	bar
JET_T	STREAM	JET	T	419.00	3.60	F
KERO_D60	STREAM	KERO	COLDDENS	0.88870	0.500E-03	D60/60
KERO_D86V0	STREAM	KERO	D86V000	431.60	5.40	F
KERO_D86V10	STREAM	KERO	D86V010	451.40	5.40	F
KERO_D86V100	STREAM	KERO	D86V100	509.00	5.40	F
KERO_D86V30	STREAM	KERO	D86V030	458.60	5.40	F
KERO_D86V5	STREAM	KERO	D86V005	447.80	5.40	F
KERO_D86V50	STREAM	KERO	D86V050	465.80	5.40	F
KERO_D86V70	STREAM	KERO	D86V070	482.00	5.40	F
KERO_D86V90	STREAM	KERO	D86V090	491.00	5.40	F
KERO_D86V95	STREAM	KERO	D86V095	500.00	5.40	F
KERO_MASSF	STREAM	KERO	MASSF	17.700	-5.000	t/h
KERO_P	STREAM	KERO	P	3.7500	-5.000	bar
KERO_T	STREAM	KERO	T	482.00	9.00	F
NAPHTHA_D60	STREAM	NAPHTHA	COLDDENS	0.77200	0.500E-03	D60/60
NAPHTHA_D86V0	STREAM	NAPHTHA	D86V000	109.40	5.40	F
NAPHTHA_D86V10	STREAM	NAPHTHA	D86V010	158.00	5.40	F
NAPHTHA_D86V100	STREAM	NAPHTHA	D86V100	311.00	5.40	F
NAPHTHA_D86V30	STREAM	NAPHTHA	D86V030	167.00	5.40	F
NAPHTHA_D86V5	STREAM	NAPHTHA	D86V005	152.60	5.40	F
NAPHTHA_D86V50	STREAM	NAPHTHA	D86V050	194.00	5.40	F
NAPHTHA_D86V70	STREAM	NAPHTHA	D86V070	230.00	5.40	F
NAPHTHA_D86V90	STREAM	NAPHTHA	D86V090	280.40	5.40	F
NAPHTHA_D86V95	STREAM	NAPHTHA	D86V095	302.00	5.40	F
NAPHTHA_MASSF	STREAM	NAPHTHA	MASSF	82.900	-5.000	t/h
NAPHTHA_P	STREAM	NAPHTHA	P	1.5000	-5.000	bar
NAPHTHA_T	STREAM	NAPHTHA	T	302.72	9.00	F

NAPHTHA_WF22MC3	STREAM	NAPHTHA	WF22MC3	1.4000	-5.000	%
NAPHTHA_WFC3	STREAM	NAPHTHA	WFC3	0.50000	-5.000	%
NAPHTHA_WFI-C4	STREAM	NAPHTHA	WFI-C4	1.5000	-5.000	%
NAPHTHA_WFN-C4	STREAM	NAPHTHA	WFN-C4	1.3000	-5.000	%
NAPHTHA_WFN-C5	STREAM	NAPHTHA	WFN-C5	2.0000	-5.000	%
RESIDU_D60	STREAM	RESIDU	COLDDENS	1.0233	0.500E-03	D60/60
RESIDU_D86V10	STREAM	RESIDU	D86V010	683.60	5.40	F
RESIDU_D86V20	STREAM	RESIDU	D86V020	771.80	5.40	F
RESIDU_D86V30	STREAM	RESIDU	D86V030	825.80	5.40	F
RESIDU_D86V40	STREAM	RESIDU	D86V040	885.20	5.40	F
RESIDU_D86V5	STREAM	RESIDU	D86V005	644.00	5.40	F
RESIDU_D86V50	STREAM	RESIDU	D86V050	946.40	5.40	F
RESIDU_D86V60	STREAM	RESIDU	D86V060	1009.4	5.40	F
RESIDU_D86V62	STREAM	RESIDU	D86V062	1020.2	5.40	F
RESIDU_MASSF	STREAM	RESIDU	MASSF	271.70	-5.000	t/h
RESIDU_P	STREAM	RESIDU	P	4.0400	-5.000	bar
RESIDU_T	STREAM	RESIDU	T	690.80	3.60	F

TABLE 11 - MEASUREMENTS FOR CRUV AL EXAMPLE

The results are (file r6v):

Number of tags	85
Number of equations	147
Number of unmeasured variables	80
Number of measured variables	132
Number of variables considered constant	123
Obvious number of redundancies	67
Total number of redundancies	67
Number of trivial redundancies	0

Validation and reconciliation results

Objective function	45,4676
Chi-square test value	87.1100
Sum of squares of residues	0.636022E-008
Number of active bounds	0
Number of variables close to bounds	0
Number of eliminated tags	0
Number of flagged tags	7
Number of filtered tags	0

TAGS IN THE MODEL

Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
CRUDE_D60	0.93180	0.500E-03	0.93176	0.490E-03	0.01	D60/60
CRUDE_MASSF	567.00	5.00 %	568.37	2.12 %	0.00	t/h
CRUDE_P	4.0000	5.00 %	4.0000	5.00 %	0.00	bar
CRUDE_T	703.40	3.60	703.40	3.60	0.00	F
DIESEL_D60	0.95000	0.500E-03	0.94996	0.382E-03	0.00	D60/60
DIESEL_D86V0	501.80	5.40	501.58	5.13	0.00	F
DIESEL_D86V10	604.40	5.40	601.67	3.91	0.26	F
DIESEL_D86V100	770.00	5.40	767.59	3.35	0.20	F
DIESEL_D86V30	636.80	5.40	638.79	3.86	0.14	F
DIESEL_D86V5	584.60	5.40	583.36	4.61	0.05	F
DIESEL_D86V50	671.00	5.40	667.21	2.41	0.49	F
DIESEL_D86V70	698.00	5.40	701.98	3.23	0.54	F
DIESEL_D86V90	739.40	5.40	742.50	2.69	0.33	F
DIESEL_D86V95	761.00	5.40	759.03	3.10	0.13	F
DIESEL_MASSF	51.100	5.00 %	51.120	4.98 %	0.00	t/h
DIESEL_P	3.9400	5.00 %	3.9400	5.00 %	0.00	bar
DIESEL_T	635.00	3.60	635.00	3.60	0.00	F

GASOIL_D60	0.90820	0.500E-03	0.90817	0.397E-03	0.00	D60/60
GASOIL_D86V0	338.00	5.40	337.84	5.32	0.00	F
GASOIL_D86V10	474.80	5.40	478.71	1.86	0.52	F
GASOIL_D86V100	626.00	5.40	619.67	3.37	1.37	F
GASOIL_D86V30	514.40	5.40	514.09	4.01	0.00	F
GASOIL_D86V5	458.60	5.40	455.27	5.00	0.38	F
GASOIL_D86V50	532.40	5.40	530.48	1.97	0.13	F
GASOIL_D86V70	555.80	5.40	558.61	3.26	0.27	F
GASOIL_D86V90	593.60	5.40	595.60	2.70	0.14	F
GASOIL_D86V95	608.00	5.40	611.46	3.12	0.41	F
GASOIL_MASSF	69.400	5.00 %	69.304	4.96 %	0.00	t/h
GASOIL_P	3.8400	5.00 %	3.8400	5.00 %	0.00	bar
GASOIL_T	552.20	3.60	552.20	3.60	0.00	F
JET_D60	0.85710	0.500E-03	0.85703	0.414E-03	0.02	D60/60
JET_D86V0	291.20	5.40	291.61	5.08	0.01	F
JET_D86V10	336.20	5.40	335.44	1.57	0.02	F
JET_D86V30	354.20	5.40	355.17	2.74	0.03	F
JET_D86V5	327.20	5.40	325.34	3.67	0.12	F
JET_D86V50	374.00	5.40	374.23	3.09	0.00	F
JET_D86V70	392.00	5.40	392.55	2.85	0.01	F
JET_D86V90	435.20	5.40	436.37	4.00	0.05	F
JET_D86V95	458.60	5.40	460.32	3.76	0.10	F
JET_MASSF	73.100	5.00 %	72.793	4.85 %	0.01	t/h
JET_P	3.6500	5.00 %	3.6500	5.00 %	0.00	bar
JET_T	419.00	3.60	419.00	3.60	0.00	F
KERO_D60	0.88870	0.500E-03	0.88834	0.326E-03	0.53	D60/60
KERO_D86V0	* 431.60	5.40	447.76	1.78	8.95	F
KERO_D86V10	451.40	5.40	449.45	1.67	0.13	F
KERO_D86V100	509.00	5.40	507.30	2.98	0.10	F
KERO_D86V30	458.60	5.40	456.75	1.38	0.12	F
KERO_D86V5	447.80	5.40	448.92	1.71	0.04	F
KERO_D86V50	465.80	5.40	467.53	1.34	0.10	F
KERO_D86V70	482.00	5.40	480.26	1.78	0.10	F
KERO_D86V90	491.00	5.40	492.65	2.53	0.09	F
KERO_D86V95	500.00	5.40	502.30	2.82	0.18	F
KERO_MASSF	17.700	5.00 %	17.689	5.00 %	0.00	t/h
KERO_P	3.7500	5.00 %	3.7500	5.00 %	0.00	bar
KERO_T	482.00	9.00	482.00	9.00	0.00	F
NAPHTHA_D60	0.77200	0.500E-03	0.77288	0.425E-03	3.13	D60/60
NAPHTHA_D86V0	109.40	5.40	110.88	5.39	0.07	F
NAPHTHA_D86V10	158.00	5.40	159.45	1.60	0.07	F
NAPHTHA_D86V100	311.00	5.40	312.04	3.92	0.04	F
NAPHTHA_D86V30	167.00	5.40	163.67	2.92	0.38	F
NAPHTHA_D86V5	152.60	5.40	159.17	1.56	1.48	F
NAPHTHA_D86V50	194.00	5.40	186.57	3.52	1.90	F
NAPHTHA_D86V70	230.00	5.40	221.41	3.87	2.53	F
NAPHTHA_D86V90	280.40	5.40	277.37	3.64	0.31	F
NAPHTHA_D86V95	302.00	5.40	300.21	3.12	0.11	F
NAPHTHA_MASSF	82.900	5.00 %	82.009	4.03 %	0.05	t/h
NAPHTHA_P	1.5000	5.00 %	1.5000	5.00 %	0.00	bar
NAPHTHA_T	302.72	9.00	302.72	9.00	0.00	F
NAPHTHA_WF22MC3	1.4000	5.00 %	1.4428	4.77 %	0.37	%
NAPHTHA_WFC3	0.50000	5.00 %	0.50919	4.88 %	0.14	%
NAPHTHA_WFI-C4	1.5000	5.00 %	1.5605	4.67 %	0.65	%
NAPHTHA_WFN-C4	1.3000	5.00 %	1.3399	4.77 %	0.38	%
NAPHTHA_WFN-C5	2.0000	5.00 %	2.0877	4.70 %	0.77	%
RESIDU_D60	* 1.0233	0.500E-03	1.0243	0.372E-03	4.22	D60/60
RESIDU_D86V10	683.60	5.40	677.37	4.59	1.33	F
RESIDU_D86V20	771.80	5.40	769.47	4.83	0.19	F
RESIDU_D86V30	825.80	5.40	821.21	4.33	0.72	F
RESIDU_D86V40	885.20	5.40	886.92	4.24	0.10	F
RESIDU_D86V5	644.00	5.40	649.09	4.24	0.89	F
RESIDU_D86V50	* 946.40	5.40	933.81	2.59	5.43	F
RESIDU_D86V60	1009.4	5.40	1000.1	3.91	2.98	F
RESIDU_D86V62	1020.2	5.40	1025.7	3.64	1.05	F
RESIDU_MASSF	271.70	5.00 %	275.45	2.81 %	0.08	t/h
RESIDU_P	4.0400	5.00 %	4.0400	5.00 %	0.00	bar
RESIDU_T	690.80	3.60	690.80	3.60	0.00	F

TABLE 12 - RESULTS OF CRUVAL EXAMPLE

SECTION 12: CUTOFF: OPTIONS FOR GENERIC CUTOFF

PURPOSE

The **CUTOFF** unit is a mathematical unit used to define the threshold values for the generic **CUTOFF** on various types of measurements. Keeping the default value of zero for the **CUTOFF** indicates no cut-off will be performed.

Any stream on which a measurement is defined with a measured value below the **CUTOFF** value is automatically set OFF.

Attention: the **CUTOFF** value defined on a particular tag overrides the generic value provided here.

SYMBOLS



FIGURE 17 - SYMBOLS FOR UNIT CUTOFF

CONNECTION RULES

No connections are allowed on a **CUTOFF** unit.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Mode
T	Parameters	Input	Threshold for Temperature
P	Parameters	Input	Threshold for Pressure
MIXF	Parameters	Input	Threshold for Mixture fraction
MOLF	Parameters	Input	Threshold for Molar flow rate
MASSF	Parameters	Input	Threshold for Mass flow rate
VOLF	Parameters	Input	Threshold for Volumetric flow rate
COLDVOL	Parameters	Input	Threshold for Volumetric flow rate at 15 C

F			
CONVVOL F	Parameters	Input	Threshold for Conventional Volumetric flow rate

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	CUTOFF not used	CUTOFF is not used
1	CUTOFF used	CUTOFF is used

T (Parameters)

Threshold for Temperature (type: Input).

P (Parameters)

Threshold for Pressure (type: Input).

MIXF (Parameters)

Threshold for Mixture fraction (type: Input).

MOLF (Parameters)

Threshold for Molar flow rate (type: Input).

MASSF (Parameters)

Threshold for Mass flow rate (type: Input).

VOLF (Parameters)

Threshold for Volumetric flow (type: Input).

COLDVOLF (Parameters)

Threshold for Volumetric flow rate at 15 C (type: Input).

CONVVOLF (Parameters)

Threshold for Conventional Volumetric flow rate (type: Input).

SECTION 13: CUTVAL: TEAR UNIT FOR LOOPS

PURPOSE

The CUTVAL unit allows breaking the material or pressure loops that can appear when dealing with closed cycles like for example the steam/water cycle in a power plant or a refrigeration cycle.

This type of unit can also be used to link a part of a model with compound balances to a part where only total mass balances are carried out.

SYMBOLS

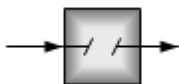


FIGURE 18 - SYMBOLS FOR UNIT CUTVAL

CONNECTION RULES

INLET STREAMS

- 1 material stream

OUTLET STREAMS

- 1 material stream

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
MODECOM P	General	Input	Composition

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	Thermal Balance	Thermal balance of the unit is considered.
1	No Thermal Balance	No thermal balance generated for the unit.

MODEDP (Parameters)

Pressure equation (type: Input)

0	None	No pressure drop equation will be generated
1	Equality	An equality for pressure between inlet and outlet will be generated

MODECOMP (Parameters)

Composition (type: Input)

0	Compounds compatibility is checked	A check that compounds in inlet and outlet streams are the same will occur
1	Compounds compatibility is NOT checked	No check occurs

In the first case, any compound not present in the inlet or outlet stream - because of a measurement or because it does not appear in its Thermod - is also set to zero in the other stream, in a similar way to other VALI unit models. In the second case, no check is made, which means that some compounds may appear or disappear through the unit.

This is useful when connecting two parts of a model with a different modeling approach (total mass balance versus compound balance). It is up to the user to add (for example through a Flex code) the equations linking the two parts (e.g. equality of total mass flow rates).

EXAMPLE : CLOSED LOOP

The goal will be to generate a closed loop by connecting black-boxes together (B-1, B-2, B-3). The loop is then broken by introducing the CUTVAL unit C-1.

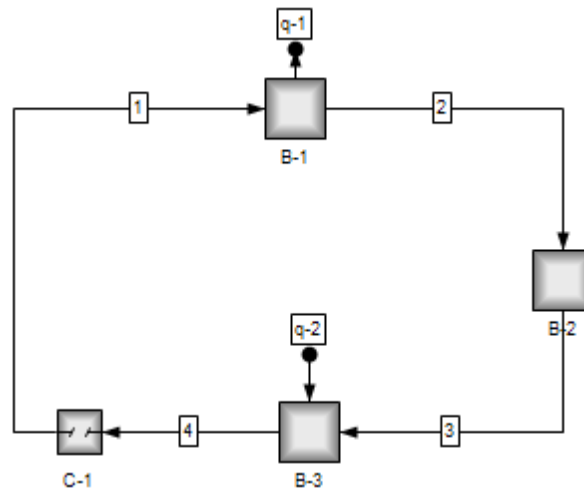


FIGURE 19 - CUTVAL EXAMPLE

There are 4 UNITS:

- B-1:
 - type=BBXVAL
 - MODE='All Equations'
 - MODETOEQ='None'
 - MODDP1='DP variable'
- B-2:
 - type=BBXVAL
 - MODE='All Equations'
 - MODETOEQ='None'
 - MODDP1='DP variable'
- B-3:
 - type=BBXVAL
 - MODE='All Equations'
 - MODETOEQ='None'
 - MODDP1='DP variable'
- C-1:
 - type=CUTVAL
 - MODE='Thermal Balance'
 - MODDP='None'

6 STREAMS must be defined:

STREAMS	THERMOD	CONNECTIONS
1	WATER	from C-1 to B-1
2	WATER	from B-1 to B-2
3	WATER	from B-2 to B-3
4	WATER	from B-3 to C-1
Q-1	THERMAL STREAM	from B-1
Q-2	THERMAL STREAM	to B-3

The Thermod WATER uses a single compound: H₂O.

The list of measurements is given in table below.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
1_MASSF	STREAM	1	MASSF	10.000	-5.000	kg/s
1_P	STREAM	1	P	5.0000	-3.000	bar
1_T	STREAM	1	T	29.000	1.00	C
2_MASSF	STREAM	2	MASSF	11.000	-5.000	kg/s
2_P	STREAM	2	P	5.0000	-3.000	bar
2_T	STREAM	2	T	10.000	1.00	C
4_T	STREAM	4	T	30.000	1.00	C
B-1_DP1	UNIT	B-1	DP1	0.0000	CST	bar
B-2_DP1	UNIT	B-2	DP1	0.0000	CST	bar
B-3_DP1	UNIT	B-3	DP1	0.0000	CST	bar

TABLE 13 - MEASUREMENTS FOR CUTVAL EXAMPLE

The results are (file r6v):

Number of tags	10					
Number of equations	11					
Number of unmeasured variables	9					
Number of measured variables	6					
Number of variables considered constant	3					
Obvious number of redundancies	2					
Total number of redundancies	3					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	2.30995					
Chi-square test value	7.81000					
Sum of squares of residues	0.118713E-019					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
1_MASSF	10.000	5.00 %	10.452	3.54 %	0.82	kg/s
1_P	5.0000	3.00 %	5.0000	2.12 %	0.00	bar
1_T	29.000	1.00	29.500	0.707	0.25	C
2_MASSF	11.000	5.00 %	10.452	3.54 %	0.99	kg/s
2_P	5.0000	3.00 %	5.0000	2.12 %	0.00	bar
2_T	10.000	1.00	10.000	1.00	0.00	C
4_T	30.000	1.00	29.500	0.707	0.25	C
B-1_DP1	0.0000	CST	0.0000			bar
B-2_DP1	0.0000	CST	0.0000			bar
B-3_DP1	0.0000	CST	0.0000			bar

TABLE 14 - RESULTS OF CUTVAL EXAMPLE

SECTION 14: DISVAL: TRAY-BY-TRAY DISTILLATION COLUMN

PURPOSE

This type of UNIT allows you to model a distillation (or absorber) column based on a tray by tray Liquid/Vapor equilibrium. Non equilibrium can be considered through the Murphree efficiency.

This new unit is delivered as is and requires a reasonable estimate of the composition and state of the outlet streams in order to get convergence of the calculations.

Care should also be taken that the thermodynamic model is adequate to represent with sufficient accuracy the Liquid/Vapor equilibrium of the corresponding mixtures. Using an inadequate thermodynamic model may strongly bias the results.

Care should also be taken with compounds that do not appear in the column feed (but are defined in the thermod). These compounds should also be set to zero in the other inlet streams of the column (reflux and reboiled stream) to avoid convergence problems.

SYMBOLS

Column

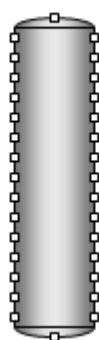


FIGURE 20 - SYMBOLS FOR UNIT DISVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 10
- Type: material (at least one), thermal

OUTLET STREAMS

- Maximum number: 10
- Type: material (at least two), thermal

The STREAMS can be of L, V or LV state. It is however recommended to use the L state for each outlet stream that is specified as liquid, and the V state for each outlet stream that is specified as vapor.

This will lead to a much simpler equation system and, thus, to better convergence.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
INIMOD	General	Input	Initialization Mode
EFFIMOD	General	Input	Efficiency Mode
NBRPLT	Parameters	Input	Number of plates
EFFIC	Parameters	Measurable	Plate Efficiency
DPPLATE	Parameters	Measurable	Plate Pressure drop
DPTOTAL	Parameters	Measurable	Total Pressure drop
PLT.IN	Parameters	Input	Plate number
PLT.OU	Parameters	Input	Plate number
TYP.OU	Parameters	Input	Stream Type
DP.IN	Parameters	Measurable	Pressure drop at entrance

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	All equations	Mass, energy balances and L/V equilibrium of each plate are taken into account
1	Mass Balance only	Only global mass balances of the column are taken into account
2	Mass and Energy Balances	Global mass and energy balances of the column are taken into account

INIMOD (General)

Initialization Mode (type:Input).

0	Top/bottom linear profile	the column profile is initialized by a simple interpolation between the top and bottom outlet streams. Input and side draw streams are not taken into account
1	Top/bottom profile + flash vapf = 10..90%	atsame as 0, but the profile is further corrected by a flash calculation made on basis of the plate pressure and an increasing vapor fraction (from 10 to 90 %)
2	Constant profile between streams	allconstant profile between two successive inlet and/or outlet streams
3	Linear profile between streams	allarithmetic interpolation between two successive inlet and/or outlet streams
4	Geometric profile between streams	allgeometric interpolation between two successive inlet and/or outlet streams
10	From file (Simu profile)	the column profile is initialized from a file (generated by SIMU software or by VALI in a preceding run)

The column profile (temperature, pressure and composition) can be initialized from the inlet and/or outlet streams, with various correction and interpolation options, or from a profile saved in a file.

EFFIMOD (General)

Efficiency Mode (type: Input).

0	None	the plates are assumed to be at L/V equilibrium
1	Murphree's Efficiency	the plates are assumed to be non-equilibrated on basis of Murphree's Efficiency

Murphree efficiency is expressed as:

$$E_i = \frac{y_i - y_{i+1}}{K_i \cdot x_i - y_{i+1}}$$

where:

- y_i and y_{i+1} : the vapor composition on plates i and $i+1$
- x_i : the vapor composition on plates i
- K_i : the K value on plate i

NBRPLT (Parameters)

Number of plates (type: Input).

The plates are numbered from the top of the column.

EFFIC (Parameters)

Plate Efficiency (type: Measurable).

The plate efficiency defines the distance

This parameter is only used if the efficiency mode is set to 1 (Murphree's Efficiency).

DPPLATE (Parameters)

Plate Pressure drop (type: Measurable).

Pressure drop per plate.

DPTOTAL (Parameters)

Total Pressure drop (type: Measurable).

Total pressure drop of the column. The variable is only created when all pressure drops are considered as variables (i.e. the parameter PRESMODE of unit OPTVAL is set to the default value of 2).

PLT.IN (Parameters)

Plate number (type: Input).

Indicates the plate destination of the inlet STREAMS. Plate 1 designates the top of the column, while the special value 999 indicates the last (bottom) plate.

PLT.OU (Parameters)

Plate number (type: Input).

Indicates the plate destination of the outlet STREAMS. Plate 1 designates the top of the column, while the special value 999 indicates the last (bottom) plate.

TYP.OU (Parameters)

Stream Type (type: Input).

0	Not Defined	Undefined status; needs to be changed
1	Liq Side Draw	Liquid Side Draw
2	Vap Side Draw	Vapor Side Draw
3	Liquid	Liquid (usually the bottom stream)
4	Vapor	Vapor (usually the top stream)
5	Thermal	Thermal stream
6	Water Draw	Water Draw-off

Indicates the type of each outlet stream.

Attention: When connecting a new stream to a DISVAL unit, the corresponding parameter TYP.OU is set as "Not defined" and must be updated by the user.

DP.IN (Parameters)

Pressure drop at entrance (type: Measurable).

For each inlet stream a pressure drop can be specified between the stream pressure and the pressure of the corresponding plate.

EXAMPLE : METHYL ACETATE/METHANOL DISTILLATION

The example is in the directory "\examples\disval" and shows the interconnection of two columns at different pressure enabling to break the azeotrope between methyl acetate and methanol.

SECTION 15: DPVAL: PRESSURE DROP (VALVE)

PURPOSE

The **DPVAL** is used to model pressure drop through lines and valves. The pressure drop can be linked to the stream rate and conditions as well as to the valve stem position.

SYMBOLS

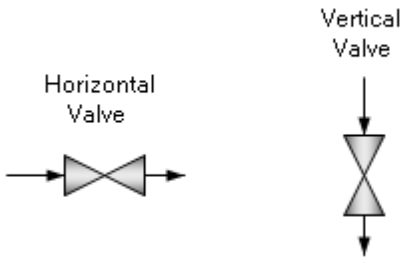


FIGURE 21 - SYMBOLS FOR UNIT DPVAL

CONNECTION RULES

INLET STREAMS

- 1 material STREAM

OUTLET STREAMS

- 1 material STREAM

Both streams must use the same THERMOD.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
METHODDP	General	Input	Multiphase flow method for DP
DP	Parameters	Measurable	Pressure drop

NAME	TAB	TYPE	DESCRIPTION
DPX	Parameters	Input	Nominal pressure drop
MASSF	Parameters	Input	Nominal flow rate
CFDP	Parameters	Measurable	DP correction factor
K	Parameters	Measurable	Pressure drop factor
EXP	Parameters	Input	Exponent of the mass flow rate
X	Parameters	Measurable	Valve stem position
CVMAX	Parameters	Measurable	Maximum value of the valve coefficient
RANGE	Parameters	Measurable	Valve range
CVFIT	Others	Input	Coefficients for the sigmoid Cv curve
AFIT	Others	Input	Coefficients for the sigmoid port area curve

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Mass and Energy Balance	Mass and energy balances are taken into account
1	Mass Balance only	Only mass balances are taken into account

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * \left(\frac{MASSF}{MASSF_{X}} \right)^2$$

		where:
		■ CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%) ■ DPX: the nominal pressure drop ■ MASSFX: the nominal flow rate ■ MASSF: the actual flow rate
3	K factor	a pressure drop equation will be generated and the pressure drop will be a function of the flow conditions according to: $DP = CFDP * K * \frac{MASSF^{EXP} * P_{IN} * VW}{(P_{IN} + P_{OUT})}$ where: ■ VW: the inlet stream volumetric weight (m³/kg) ■ K: a measurable factor ■ EXP: a constant value This equation has been developed for compressible fluids. However, it can also be used for liquids. In any case, the flow regime must remain the same.
4	Cv linear	$C_V = X * (CVMAX - CVMIN) + CVMIN$
5	Cv square root	$C_V = \sqrt{X} * CVMAX$
6	Cv equal percentage	$C_V = RANGE^{X-1} * CVMAX$ If $RANGE < 1e-10$, RANGE is set to 1e-10.
7	Cv hyperbolic	$C_V = \frac{CVMAX}{RANGE - (RANGE - 1) * X}$
8	Cv Flex	Cv will be given through a Flex equation written in the DPVAL "Flex code" tab
9	Cv correlation	Cv will be a sigmoid shaped function of stem travel percentage; the equation parameters have to be provided to Vali by introducing them into the CVFIT column found under the "Others" tab in DPVAL $C_V = Ymin + (Ymax - Ymin) * \frac{(YY - YY0)}{(YY1 - YY0)}$ where:

$$XX = \frac{X - X_{min}}{X_{max} - X_{min}}$$

$$YY = \tanh(YA * XX - YB) + YE * \tanh(YV * XX - YD) + YF * XX^2 + YG * XX^3 + YH * XX^4$$

$$YY0 = \tanh(-YB) + YE * \tanh(-YD)$$

$$YY1 = \tanh(YA - YB) + YE * \tanh(YC - YD) + YF + YG + YH$$

$$X_{min} = 0.0$$

$$X_{max} = 1.0$$

$$Y_{min} = \frac{CV_{MIN}}{CONVCV}$$

$$Y_{max} = \frac{CV_{MAX}}{CONVCV}$$

$$CONVCV = 2.4027157096e - 5$$

$$YA = CVFIT1$$

$$YB = CVFIT2$$

$$YC = CVFIT3$$

$$YD = CVFIT4$$

$$YE = CVFIT5$$

$$YF = CVFIT6$$

$$YG = CVFIT7$$

$$YH = CVFIT8$$

10 Area Flex Area will be given through a Flex equation written in the DPVAL "Flex code" tab

11 Area correlation Port area will be a sigmoid shaped function of stem travel percentage; the equation parameters have to be provided to Vali by introducing them into the AFIT column found under the "Others" tab in the DPVAL

$$Area = Y_{min} + (Y_{max} - Y_{min}) * \frac{(YY - YY0)}{(YY1 - YY0)}$$

where the variables are the same than the ones defined for the "9 - Cv Correlation" Mode, except:

- $Y_{min} = AREA_{min}$
- $Y_{max} = AREA_{max}$

For all these modes, a pressure drop equation will be generated and the pressure drop will be a function of the flow conditions according to an equation having the following form:

$$DP = \frac{10^{-5} * MOLF^2 * MV * MOLW}{Cv_{max}^2 * Cv^2}$$

where:

- MOLF: the stream molar flowrate (kmol/s)
- MV: the stream molar volume (m³/kmol)
- MOLV: the mean molecular weight of the mixture (kg/kmol)

- Cv: related to the dimensionless stem position X
- X: the normalized valve stem position ($X=X_{\text{actual}}/X_{\text{max}}$)
- Cvmax: the Cv of valve at maximum opening and is related to the conventional Cv (expressed in US terms) by: $Cv=2.40273 \cdot 10^{-5} \cdot Cv_{\text{US}}$

METHODDP (General)

Multiphase flow method for DP (type: Input).

This mode models pressure drop for multiphase water/oil/gas mixes flowing across choke valves, found in upstream applications. It is intended to be applied exclusively to WLV mixes. The multiphase correlations can only be used with MODEDP Area Flex or Area Correlation if port area is to be defined as a function of stem position.

0	None	No pressure drop equation will be generated
1	Sachdeva	A pressure drop equation will be generated. The pressure drop will be a function of the mass flow rate according to the following equation:

$$M_2 = A c_d \left(288 g_c p_1 \rho_{m2}^2 \left(\frac{(1-x_1)(1-y)}{\rho_L} + \frac{x_1 k}{k-1} (V_{G1} - y V_{G2}) \right) \right)^{0.5}$$

where:

- M_2 : mass flow rate (lbm/s)
- A: downstream hydraulic flow area (ft² - port area at a given opening)
- c_d : valve discharge coefficient (valve parameter)
- g_c : acceleration due to gravity (32.17 ft²/s)
- p_1 : pressure upstream of valve (psi)
- ρ_{m2} : mixture density downstream (lbm/ft³)
- x_1 : gas in-situ quality at upstream (mass fraction)
- y: downstream to upstream pressure ratio
- k: specific heat ratio (C_{pg}/C_{vg})
- V_{G1} : gas specific volume upstream (ft³/lbm)
- V_{G2} : gas specific volume downstream (ft³/lbm)
- ρ_L : liquid density (lbm/ft³)

A critical pressure ratio equation, y_c , will be generated:

$$y_c = \left(\frac{\frac{k}{k-1} + \frac{(1-x_1)V_L(1-y_c)}{x_1V_{G1}}}{\frac{k}{k-1} + \frac{n}{2} + \frac{n(1-x_1)V_L}{x_1V_{G2c}} + \frac{n}{2} \left(\frac{(1-x_1)V_L}{x_1V_{G2c}} \right)^2} \right)^{\frac{k}{k-1}}$$

where:

- y_c : critical downstream to upstream pressure ratio
- n : polytropic exponent for the gas
- V_L : liquid specific volume upstream (ft³/lbm)
- V_{G2c} : critical gas specific volume downstream (ft³/lbm)

The calculated value of y_c will be compared with the actual pressure drop value, y , to check for sonic flow conditions. If $y < y_c$ gas flow rate reaches its maximum value and y_c will be used in the DP=F(MASSF) equation.

2 Perkins

A set of equations will be generated and the pressure drop will be a function of the mass flow rate.

The downstream mixture fluid velocity, u_2 , can be calculated by:

$$u_2 = \sqrt{\frac{288g_c \left\{ \lambda p_1 V_{G1} \left[1 - y^{(n-1)/n} \right] + \left(\frac{x_o}{\rho_o} + \frac{x_w}{\rho_w} \right) p_1 (1-y) \right\}}{1 - \left(\frac{A_2}{A_1} \right)^2 \left[\frac{(x_g + \alpha_1)}{(x_g y^{-1/n} + \alpha_1)} \right]^2}}$$

where:

- g_c : acceleration due to gravity (32.17 ft²/s)
- λ : group parameter
- V_{G1} : gas specific volume upstream (ft³/lbm)
- y : downstream to upstream pressure ratio
- n : polytropic exponent for the gas

- x_o : oil in-situ quality at upstream (mass fraction)
- x_w : water in-situ quality at upstream (mass fraction)
- ρ_o : oil density upstream (lbm/ft³)
- ρ_w : water density upstream (lbm/ft³)
- p_1 : pressure upstream of valve (psi)
- A_1 : upstream hydraulic flow area (ft²)
- A_2 : downstream hydraulic flow area (ft² - port area at a given opening)
- x_g : gas in-situ quality at upstream (mass fraction)
- α_1 : group parameter

Mass flow rate for the mixture, M_2 (lbm/s), is then given by:

$$M_2 = A_2 \rho_2 u_2$$

where:

- ρ_2 : mixture density downstream (lbm/ft³)
- u_2 : mixture fluid velocity downstream (ft/s)

with group parameters:

$$\lambda = x_g + \frac{(x_g c_{vg} + x_o c_{vo} + x_w c_{vw}) M_W}{zR} \quad \alpha_1 = \frac{1}{V_{G1}} \left(\frac{x_o}{\rho_o} + \frac{x_w}{\rho_w} \right)$$

where:

- c_{vg} : gas specific heat (BTU/lbm.R)
- c_{vo} : oil specific heat (BTU/lbm.R)
- c_{vw} : water specific heat (BTU/lbm.R)
- M_W : molecular weight of gas (lbm/mol)
- z : gas compressibility factor
- R : universal gas constant (10.73 psi.ft³/mole.R)

A critical pressure ratio equation, y_c will be generated:

$$\left[\lambda \left(-y_e^{-(n-1)/n} \right) + 2\alpha_1 (1 - y_e) \right] \left[1 - \left(\frac{A_2}{A_1} \right)^2 \left(\frac{x_g + \alpha_1}{x_g y_e^{-1/n} + \alpha_1} \right)^2 \right] + \left(\frac{x_g}{n} y_e^{-(1+n)/n} \right) + \left(\frac{A_2}{A_1} \right)^2 \left(\frac{x_g}{n} \right) \frac{(x_g + \alpha_1) y_e^{-(1+n)/n}}{(x_g y_e^{-1/n} + \alpha_1)} \right] \\ = \left[1 - \left(\frac{A_2}{A_1} \right)^2 \left(\frac{x_g + \alpha_1}{x_g y_e^{-1/n} + \alpha_1} \right)^2 \right] \left(x_g y_e^{-1/n} + \alpha_1 \right) \left[\lambda \left(\frac{n-1}{n} \right) y_e^{-1/n} + \alpha_1 \right]$$

The calculated value of y_c will be compared with the actual pressure drop value, y , to check for sonic flow conditions. If $y < y_c$ gas flow rate reaches its maximum value and y_c will be used in the $DP=F(MASSF)$ equation.

3 Sun et al.

A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to the following equation:

$$M_2 = A_2 c_d \left[\frac{288 g_c P_1 \left[V_{G1} y^{\frac{1}{n}} x_g \left(\frac{n}{n-1} \right) \left(y - y^{\frac{1}{n}} \right) + \left(\frac{x_o \rho_w + x_w \rho_o}{\rho_o \rho_w (x_o + x_w)} \right) (-x_g) (y-1) \right]}{\left(\beta^2 - 1 \right) \left[x_g \left(V_{G1} y^{\frac{1}{n}} - \left(\frac{x_o \rho_w + x_w \rho_o}{\rho_o \rho_w (x_o + x_w)} \right) \right) + \left(\frac{x_o \rho_w + x_w \rho_o}{\rho_o \rho_w (x_o + x_w)} \right)^2 \right]} \right]^{\frac{1}{2}}$$

where:

- M_2 : mass flow rate (lbm/s)
- A_2 : downstream hydraulic flow area (ft² - port area at a given opening)
- c_d : valve discharge coefficient (valve parameter)
- g_c : acceleration due to gravity (32.17 ft/s)
- P_1 : pressure upstream of valve (psi)
- V_{G1} : gas specific volume upstream (ft³/lbm)
- y : downstream to upstream pressure ratio
- n : polytropic exponent
- x_g : gas in-situ quality at upstream (mass fraction)
- x_o : oil in-situ quality at upstream (mass fraction)
- x_w : water in-situ quality at upstream (mass fraction)
- ρ_o : oil density upstream (lbm/ft³)
- ρ_w : water density upstream (lbm/ft³)
- β : upstream-downstream fluid velocity ratio

where:

$$\beta = \frac{A_2}{A_1} \left(\frac{V_{G1} x_g + \frac{x_o}{\rho_o} + \frac{x_w}{\rho_w}}{V_{G1} y^{\frac{1}{n}} x_g + \frac{x_o}{\rho_o} + \frac{x_w}{\rho_w}} \right)$$

where:

- A_1 : upstream hydraulic flow area (ft²)

A critical pressure ratio equation, y_c will be generated by:

$$V_{G1} y_c^{\frac{1}{n}} x_g \left(\frac{n}{n-1} \left[1 - \left(\frac{1}{y_c} \right)^{\frac{n-1}{n}} \right] + \left(\frac{x_o}{\rho_o} + \frac{x_w}{\rho_w} \right) \left(1 - \frac{1}{y_c} \right) \right) - \frac{n(\beta^2 - 1) \left[x_g \left(V_{G1} y_c^{\frac{1}{n}} - \left[\frac{x_o \rho_w + x_w \rho_o}{\rho_o \rho_w (x_o + x_w)} \right] \right) + \left(\frac{x_o \rho_w + x_w \rho_o}{\rho_o \rho_w (x_o + x_w)} \right) \right]^2}{2 x_g V_{G1} y_c^{\frac{1}{n}}} = 0$$

The calculated value of y_c will be compared with the actual pressure drop value, y , to check for sonic flow conditions. If $y < y_c$ gas flow rate reaches its maximum value and y_c will be used in the $DP=F(\text{MASSF})$ equation.

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{in} - P_{out}$$

This equation is not generated if $MODEDP = 0$.

where:

- P_{in} : the pressure of the inlet stream
- P_{out} : the pressure of the outlet stream

K (Parameters)

Pressure drop factor (type: Measurable).

EXP (Parameters)

Exponent of the mass flow rate (type: Input).

These two parameters are only use when MODEDP = 3 ('K factor').

X (Parameters)

Valve stem position (type: Measurable).

CVMAX (Parameters)

Maximum value of the valve coefficient (type: Measurable).

RANGE (Parameters)

Valve range (type: Measurable).

The valve range is equal to the ratio between the maximum controllable flow and the minimum controllable flow

These three last parameters are only used in some cases (depending on the value of MODEDP).

DIAM (Parameters)

Internal pipe diameter (type: Input).

This parameter is only used when METHODDP is not equal to 0.

AREAMIN (Parameters)

Minimum value for port area (type: Measurable).

This parameter is only used when METHODDP is not equal to 0.

AREAMAX (Parameters)

Maximum value for port area (type: Measurable).

This parameter is only used when METHODDP is not equal to 0.

ACTAREA (Parameters)

Actual value of port area for a given opening (type: Measurable).

This parameter is only used when METHODDP is not equal to 0.

YC (Parameters)

Critical downstream-to-upstream pressure ratio for sonic flow across the valve for WLV mixes (type: Result).

This parameter is only used when METHODDP is not equal to 0.

YP (Parameters)

Actual downstream-to-upstream pressure ratio for sonic flow across the valve for WLW mixes (type: Result).

This parameter is only used when METHODDP is not equal to 0.

Y (Parameters)

Downstream-to-upstream pressure ratio used in the selected WLW pressure drop equation (type: Result).

This parameter is only used when METHODDP is not equal to 0.

M2 (Parameters)

Mass flow rate calculated using WLW pressure drop equation (type: Result).

This parameter is only used when METHODDP is not equal to 0.

CD (Parameters)

Valve discharge coefficient (type: Measurable).

This parameter is only used when METHODDP is not equal to 0.

CVFIT (Others)

Coefficients 1 thorough 7 (in increasing order) for the sigmoid Cv curve (type: Input).

AFIT (Others)

Coefficients 1 thorough 7 (in increasing order) for the sigmoid port Area curve (type: Input).

Attention: The example on BWHVAL unit also includes a DPVAL unit.

SECTION 16: ECOVAL: BOILER ELEMENT

PURPOSE

This type of UNIT is dedicated to the detail modeling of tube layers in vertical and horizontal heat recovery steam boilers (HRSG). It includes predictions of heat transfer coefficients and pressure drop on fumes and steam side. The same UNIT type can be used for the modeling of economizers, vaporizers and super-heaters.

Attention: This type of unit is not part of the standard VALI installation. It is available as an add-in to VALI under a special license agreement.

SYMBOLS

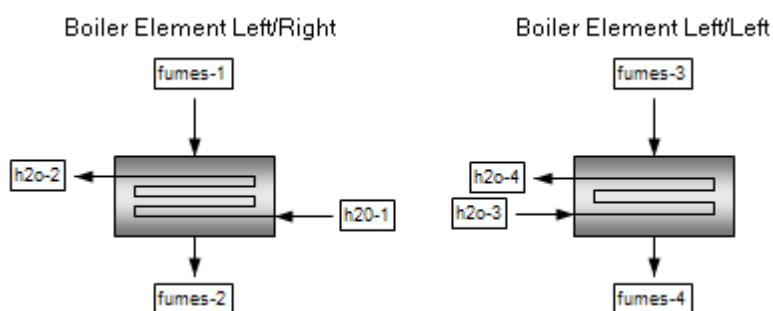


FIGURE 22 - SYMBOLS FOR UNIT ECOVAL

CONNECTION RULES

INLET STREAMS

- 2 material STREAMS: fumes in and steam/water in

OUTLET STREAMS

- 2 material STREAM: fumes out and steam/water out

THERMODS:

Corresponding inlet and outlet STREAMS must use the same THERMOD.

Fumes can go up or down (both fumes hot-spots can be either inlet or an outlet).

The water/steam comes in from the upper left side to either the down right or down left side. Mirroring the symbol will generate alternate configurations.

SECTION 17: EEXVAL: HEAT EXCHANGER

PURPOSE

This type of UNIT allows you to model the heat and mass balances of heat exchangers of various layouts. Up to ten different fluids may be considered as well as some thermal streams.

This unit does not check for temperature crossing between hot and cold streams during calculation. To prohibit this behavior use units of type HEXVAL or PRHVAL. During the report phase, temperature crossing will be detected and reported when the **EEXVAL** is modeling the heat exchange between 2 material streams.

SYMBOLS

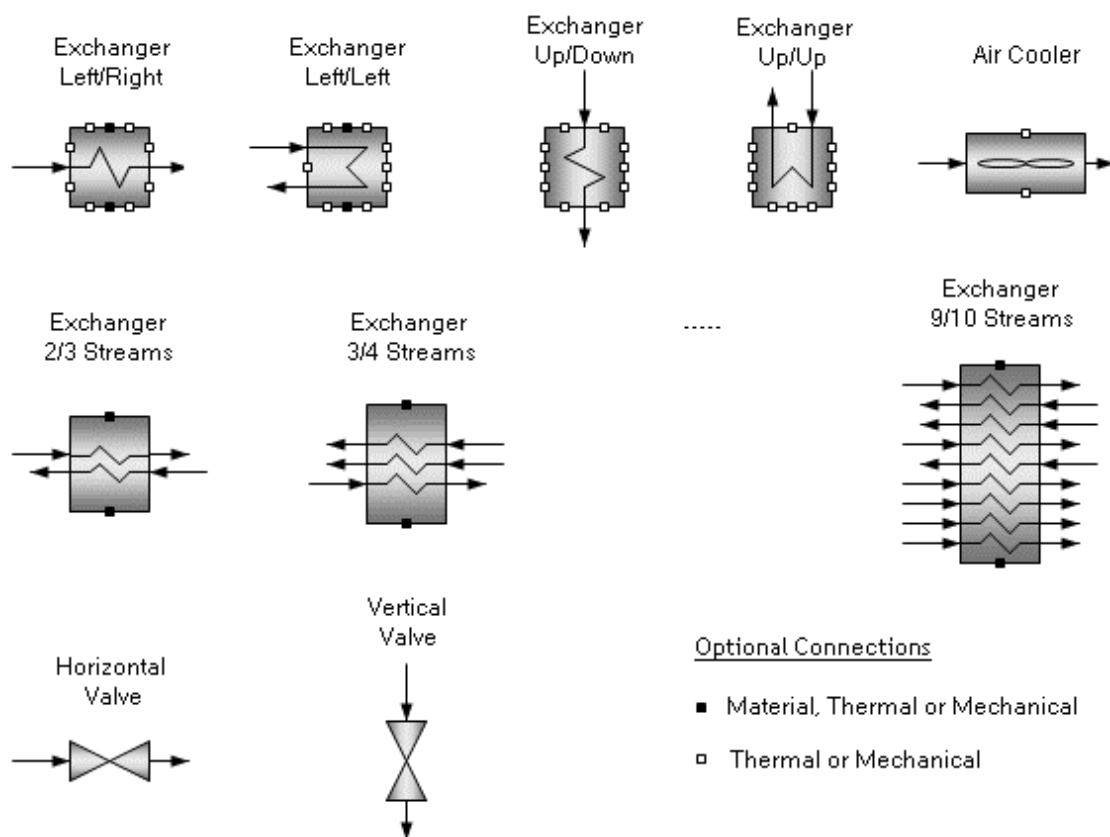


FIGURE 23 - SYMBOLS FOR UNIT EEXVAL

CONNECTION RULES

The streams indicated in above figure must be material. Corresponding inlet/outlet streams must use the same THERMOD. The unused hot-spots are optional.

The symbols of the second row allow you to handle any number of streams up to 10. The upper and lower hot-spots can be used for any type of stream. The horizontal lanes can only be used for material streams. The direction of these streams is free.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account (0,1)
LOAD	Parameters	Result	Load of the exchanger
UA1,UA2	Parameters	Result	Heat transfer coefficients
MODEDPi	Parameters	Input	Pressure equation
DPi	Parameters	Measurable	Pressure drop
DPXi	Parameters	Input	Nominal pressure drop
MASSFxi	Parameters	Input	Nominal flow rate
CFDPi	Parameters	Measurable	DP correction factor

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Mass and Energy Balance	Mass and energy balances are taken into account
1	Mass Balance only	Only mass balances are taken into account

No equations are considered for the heat exchanger performance (use [HEXVAL](#) instead).

LOAD (Parameters)

Load of the exchanger (type: Result).

This parameter shows the amount of heat transferred by the unit. It does not need to be initialized and is calculated after validation (only when the exchange concerns 2 material streams only).

UA1,UA2 (Parameters)

Heat transfer coefficients (type: Result).

- UA1: calculated on basis of counter-current flow
- UA2: calculated on basis of co-current lay-out

These values do not need to be initialized and must be calculated after validation.

MODEDP (Parameters)

Pressure equation (type: Input)

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * (MASSF / MASSFX)^2$$

where:

- CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%)
- DPX: the nominal pressure drop
- MASSFX: the nominal flow rate
- MASSF: the actual flow rate

A value of MODEDP has to be given for each outlet stream, although it is only meaningful for material streams

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{in} - P_{out}$$

This equation is not generated if MODEDP = 0.

where:

- P_{in} : the pressure of the inlet stream
- P_{out} : the pressure of the outlet stream

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when MODEDP = 2.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

EXAMPLE : HEAT EXCHANGE BETWEEN GAS AND WATER.

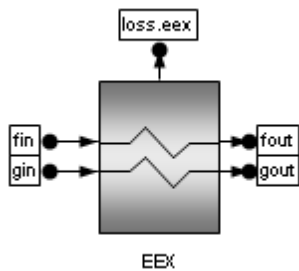


FIGURE 24 - EEXVAL EXAMPLE: GAS/WATER EXCHANGE

There is 1 UNIT :

- UNIT : EEX
 - Type=EEXVAL
 - MODE='Mass and Energy Balance'
 - MODEDP1='DP=F(MASSF)'
 - MODEDP2='DP=F(MASSF)'

5 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
FIN	FMOD	to EEX
GIN	STEAM	to EEX
FOUT	FMOD	from EEX
GOUT	STEAM	from EEX
LOSS.EEX	THERMAL STREAM	from EEX

The THERMODS are:

- FMOD: H2, N2, CO2, H2O
- STEAM: H2O

The THERMODSTEAM is readily available in the file Empty.bls. The THERMOD FMOD can easily be derived from the THERMOD FUMES.

The list of measurements is given in table below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
EEX_CFDP1	UNIT	EEX	CFDP1	1.0000	-30.00	-
EEX_CFDP2	UNIT	EEX	CFDP2	1.0000	-30.00	-
EEX_DP1	UNIT	EEX	DP1	1.0000	0.200E-01	bar
EEX_DP2	UNIT	EEX	DP2	5.0000	0.500	bar
FIN_MFCO2	STREAM	FIN	MFCO2	13.000	-5.000	%
FIN_MFH2	STREAM	FIN	MFH2	42.000	-5.000	%
FIN_MFH2O	STREAM	FIN	MFH2O	27.000	-5.000	%
FIN_MOLF	STREAM	FIN	MOLF	0.34000E+06	-5.000	Nm3/h
FIN_P	STREAM	FIN	P	30.000	-2.000	bar
FIN_T	STREAM	FIN	T	231.00	1.00	C
FOUT_P	STREAM	FOUT	P	29.000	-2.000	bar
FOUT_T	STREAM	FOUT	T	180.00	1.00	C
GIN_P	STREAM	GIN	P	125.00	-2.000	bar
GIN_T	STREAM	GIN	T	128.00	1.00	C
GOUT_MASSF	STREAM	GOUT	MASSF	79.000	-5.000	t/h
GOUT_P	STREAM	GOUT	P	120.00	-2.000	bar
GOUT_T	STREAM	GOUT	T	200.00	1.00	C
LOSS.EEX_LOAD	STREAM	LOSS.EEX	LOAD	100.00	-5.000	kW

TABLE 15 - MEASUREMENTS FOR EEXVAL EXAMPLE

The results are (file r6v):

Number of tags	18
Number of equations	14
Number of unmeasured variables	9
Number of measured variables	18
Number of variables considered constant	0
Obvious number of redundancies	5
Total number of redundancies	5
Number of trivial redundancies	0

Validation and reconciliation results

Objective function	1.02276
Chi-square test value	11.0800
Sum of squares of residues	0.309133E-011
Number of active bounds	0
Number of variables close to bounds	0
Number of eliminated tags	0
Number of flagged tags	0
Number of filtered tags	0

TAGS IN THE MODEL

Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
EEX_CFD1	1.0000	30.0 %	1.0465	9.85 %	0.02	-
EEX_CFD2	1.0000	30.0 %	1.1924	11.3 %	0.41	-
EEX_DP1	1.0000	0.200E-01	0.99980	0.200E-01	0.00	bar
EEX_DP2	5.0000	0.500	4.8716	0.462	0.07	bar
FIN_MFCO2	13.000	5.00 %	12.992	5.00 %	0.00	%
FIN_MFH2	42.000	5.00 %	41.933	4.89 %	0.00	%
FIN_MFH2O	27.000	5.00 %	26.933	4.99 %	0.00	%
FIN_MOLF	0.34000E+06	5.00 %	0.33465E+06	3.67 %	0.10	Nm3/h
FIN_P	30.000	2.00 %	29.998	1.39 %	0.00	bar
FIN_T	231.00	1.00	230.86	0.964	0.02	C
FOUT_P	29.000	2.00 %	28.998	1.44 %	0.00	bar
FOUT_T	180.00	1.00	180.14	0.964	0.02	C
GIN_P	125.00	2.00 %	124.93	1.40 %	0.00	bar
GIN_T	128.00	1.00	127.90	0.983	0.01	C
GOUT_MASSF	79.000	5.00 %	81.355	3.64 %	0.36	t/h
GOUT_P	120.00	2.00 %	120.06	1.45 %	0.00	bar
GOUT_T	200.00	1.00	200.10	0.982	0.01	C
LOSS.EEX_LOAD	100.00	5.00 %	100.03	5.00 %	0.00	kW

TABLE 16 - RESULTS OF EEXVAL EXAMPLE

SECTION 18: EFFVAL: EFFICACITY AND LOSSES

PURPOSE

The **EFFVAL** is used to model mechanical and/or thermal efficiencies and/or losses.

SYMBOLS

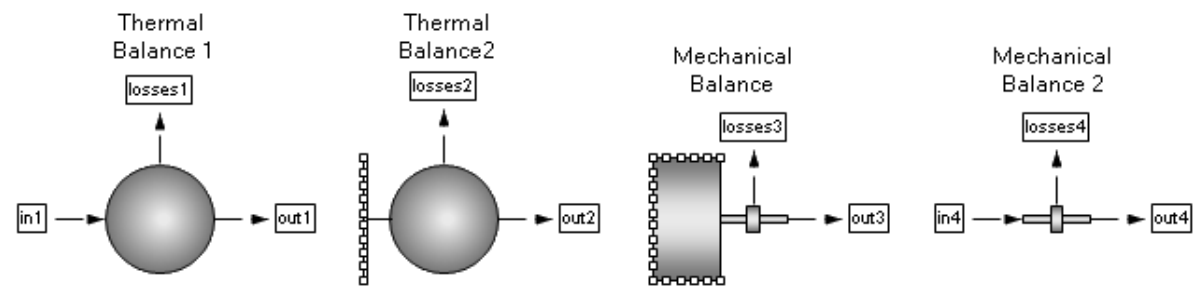


FIGURE 25 - SYMBOLS FOR UNIT EFFVAL

CONNECTION RULES

INLET STREAMS

- Maximum number : 10
- Type : mechanical, thermal

OUTLET STREAMS

- Number: 2
- Type: mechanical, thermal

The second stream corresponds to the losses.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account (0,1)
EFFIC	Parameters	Measurable	Efficiency
LOSSES	Parameters	Measurable	Relative Losses

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Default	Efficiencies and losses are taken into account
1	Mass Balance only	No equation

Effic (Parameters)

Efficiency (type: Measurable).

The efficiency is the ratio between the first outlet heat load or mechanical power and the sum of the inlet heat load and/or mechanical power.

Losses (Parameters)

Relative Losses (type: Measurable).

The relative losses are the ratio between the second outlet heat load or mechanical.

SECTION 19: EQUVAL: COMPOSITION EQUALITY

PURPOSE

The EQUVAL is used to link the composition of 2 material streams.

SYMBOLS

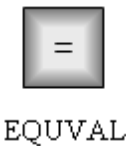


FIGURE 26 - SYMBOLS FOR UNIT EQUVAL

CONNECTION RULES

No connections are allowed on an EQUVAL.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
ALPHA	Parameters	Results	Internal parameter
>SNAME1	Others	Input	Name of first stream
>SNAME2	Others	Input	Name of second stream

UNIT PARAMETERS

>SNAME1 (Others)

Name of first stream (type: Input).

>SNAME2 (Others)

Name of second stream (type: Input).

The names of 2 streams that must have the same composition is specified in the Others page.

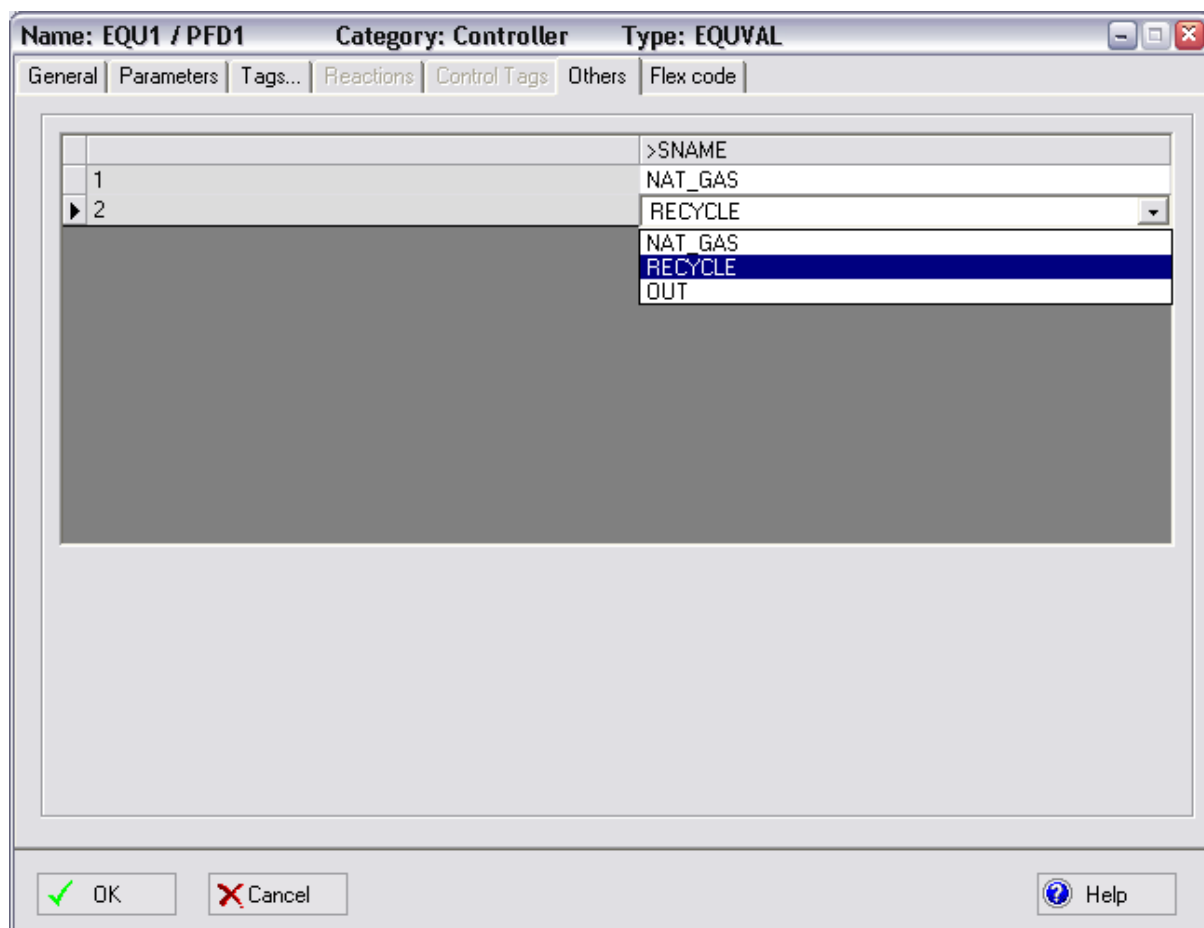


FIGURE 27 - THE OTHERS PAGE OF AN EQUVAL UNIT

EXAMPLE: COMPOSITION EQUALITY BETWEEN 2 GAS STREAMS

In this example, two gas streams are mixed. The main natural stream has a known composition but not the recycle gas stream. The two compositions are linked together through a EQUVAL unit

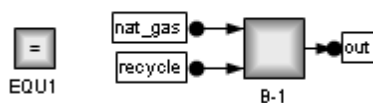


FIGURE 28 - EQUVAL EXAMPLE

There are 2 UNITS:

- B-1
 - type=BBXVAL
 - MODE='Mass and Energy Balance'
 - MODEDP='DP variable' on outlet stream
- EQU1
 - type=EQUVAL
 - pointing to NAT_GAS and RECYCLE

3 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
NAT_GAS	NG	to B-1
RECYCLE	NG	to B-1
OUT	NG	from B-1

The list of measurements is given and results are given in tables below.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
B-1_DP1	UNIT	B-1	DP1	0.0000	CST	bar
NAT_GAS_MFC1	STREAM	NAT_GAS	MFC1	80.000	-5.000	%
NAT_GAS_MFC2	STREAM	NAT_GAS	MFC2	10.000	-5.000	%
NAT_GAS_MFC3	STREAM	NAT_GAS	MFC3	5.0000	-5.000	%
NAT_GAS_MFI-C4	STREAM	NAT_GAS	MFI-C4	3.0000	-5.000	%
NAT_GAS_MFN-C4	STREAM	NAT_GAS	MFN-C4	2.0000	-5.000	%
NAT_GAS_MOLF	STREAM	NAT_GAS	MOLF	15000.	-5.000	Nm3/h
NAT_GAS_P	STREAM	NAT_GAS	P	2.0000	-3.000	bar
NAT_GAS_T	STREAM	NAT_GAS	T	50.000	1.00	C
OUT_LHVMOL	STREAM	OUT	LHVMOL	45800.	-1.000	kJ/Nm3
OUT_MOLF	STREAM	OUT	MOLF	19500.	-5.000	Nm3/h
OUT_T	STREAM	OUT	T	55.000	1.00	C
RECYCLE_MOLF	STREAM	RECYCLE	MOLF	5000.0	-5.000	Nm3/h
RECYCLE_P	STREAM	RECYCLE	P	2.5000	-3.000	bar
RECYCLE_T	STREAM	RECYCLE	T	65.000	1.00	C

TABLE 17 - MEASUREMENTS LIST FOR EQUVAL EXAMPLE

Number of tags	15					
Number of equations	24					
Number of unmeasured variables	20					
Number of measured variables	14					
Number of variables considered constant	1					
Obvious number of redundancies	4					
Total number of redundancies	4					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	1.39206					
Chi-square test value	9.49000					
Sum of squares of residues	0.166651E-011					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
B-1_DP1	0.0000	CST	0.0000			bar
NAT_GAS_MFC1	80.000	5.00 %	79.847	0.659 %	0.00	%
NAT_GAS_MFC2	10.000	5.00 %	10.080	4.75 %	0.03	%
NAT_GAS_MFC3	5.0000	5.00 %	5.0405	4.78 %	0.03	%
NAT_GAS_MFI-C4	3.0000	5.00 %	3.0218	4.82 %	0.02	%
NAT_GAS_MFN-C4	2.0000	5.00 %	2.0097	4.91 %	0.01	%
NAT_GAS_MOLF	15000.	5.00 %	14764.	4.05 %	0.10	Nm3/h
NAT_GAS_P	2.0000	3.00 %	2.0000	3.00 %	0.00	bar
NAT_GAS_T	50.000	1.00	50.533	0.817	0.28	C
OUT_LHVMOL	45800.	1.00 %	45551.	0.473 %	0.30	kJ/Nm3
OUT_MOLF	19500.	5.00 %	19771.	3.10 %	0.08	Nm3/h
OUT_T	55.000	1.00	54.281	0.628	0.52	C
RECYCLE_MOLF	5000.0	5.00 %	5007.2	4.86 %	0.00	Nm3/h
RECYCLE_P	2.5000	3.00 %	2.5000	3.00 %	0.00	bar
RECYCLE_T	65.000	1.00	65.186	0.980	0.03	C

TABLE 18 - RESULTS OF EQUVAL EXAMPLE

SECTION 20: FLXVAL: USER DEFINED EQUATIONS

PURPOSE

The FLXVAL unit allows to :

- perform advanced filtering operations (ValiFilter license required)
- add user defined constraints
- add user defined measurements
- calculate user defined functions of the validated values
- calculate results and generate reports

Using FLEX language, you can access values of streams and units and perform calculations. Thermodynamic routines can be used to calculate various thermodynamic properties and perform physical equilibrium compositions. The results of these calculations (up to 9 values) are then stored in the parameters VAR1 ... VAR9 of the unit. These parameters are dimensionless.

You can then:

- enter a "perfect" measurement (deviation=0) on the calculated value. This will define an additional constraint
- enter a measurement on the calculated value. This is then a user defined measurement
- not enter any measurement. The calculated values will simply be available in the various reports

Attention: A flex code can be associated to any equipment unit, defining additional constraints (all residuals of the corresponding equations are assumed to be zero).

SYMBOLS



FIGURE 29 - SYMBOLS FOR UNIT FLXVAL

CONNECTION RULES

No stream can be connected to a unit of type **FLXVAL**.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
NEQU	General	Input	Number of equations (1..9)
VARi	Parameters	Measurable	Results of the calculations

UNIT PARAMETERS

NEQU (General)

Number of equations (1..9) (type: Input).

NEQU is the number of equations to be handled by the module and can range from 1 to 9. When more equations are defined in the Flex program attached to the unit, only the first NEQU equations are actually used.

VARi (Parameters)

Results of the calculations (type: Measurable).

These variables will contain the results of the equations defined in the Flex program. Measurements can be defined on these variables. Note that these parameters are dimensionless.

DEFINING THE FLEX PROGRAM

The Flex code attached to a Unit of type FLXVAL can define up to 9 equations. It can also contain user-defined routines called from other Flex codes

Each equation must be defined in a separate FUNCTION (EQ1, EQ2, ...) and the result of the calculations must be stored in the variable RES. The value of RES will automatically be available in the variables VARi, of the unit (VAR1 for equation EQ1, etc.).

Typically, each FUNCTION consists in:

- a declaration section: defines Variables and Arrays that will be necessary
- selection of a Thermod: defines the thermodynamics methods to be used
- selection of variables: defines which variables will be used in the equation
- calculations: defines the calculations

Comments can be entered anywhere after the exclamation mark "!".

Flex syntax is similar to FORTRAN Code. Syntax is described in chapter "[Flex code](#)".

Example:

```
FUNCTION EQ1 ! This the header for the first equation
! Declaration
REAL ...
! Selection of thermodynamic methods
```

```

CALL SELECT('Astream')
! Selection of variables
CALL GETALL(...)
CALL GETVAR(...)
CALL GETUVA(...)
CALL GTVARTAG(...)
CALL GTTAGVAL(  )
! Calculations
...
RES = ...
END

```

EXAMPLE OF FLEX PROGRAM

In this example, steam is superheated in the radiation part of a boiler (HTX) and expanded in a turbine (TUR). Flex code is used to model the radiation heat transfer in the boiler and to link the steam rate going through the turbine to the pressure drop.

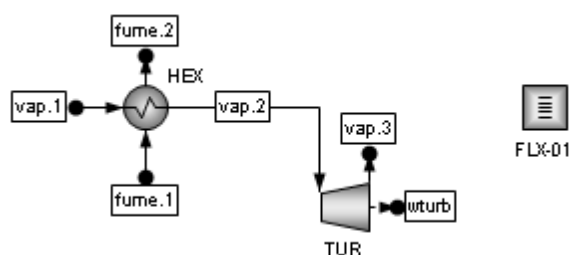


FIGURE 30 - FLXVAL EXAMPLE

There are 3 UNITS:

- HEX
 - type=HEXVAL
 - MODE='Mass and Energy Balance'
- TUR
 - type=TURVAL
 - MODE='All equations'
- FLX-01
 - type=FLXVAL
 - '2 equations'

6 STREAMS must be defined (pay attention to their state!):

STREAMS THERMOD

CONNECTIONS

VAP.1	STEAM (Liquid/Vapor)	to HTX
VAP.2	STEAM (Vapor)	from HTX to TUR
VAP.3	STEAM (Liquid/Vapor)	from TUR
FUME.1	FUMES (Vapor)	to HTX
FUME.2	FUMES (Vapor)	from HTX
WTURB	Mechanical	from TUR

The flex code introduces two constraints:

- radiation zone of the boiler

$$\text{LOAD} = 10^9 \text{ VAR1} \left(T_{\text{Fume.2}}^4 - T_{\text{Vap.2}}^4 \right)$$

- pressure drop through boiler:

$$P_{\text{Vap.2}}^2 - P_{\text{Vap.3}}^2 = \text{VAR 2 } G_{\text{Vap.2}} P_{\text{Vap.2}} V_{\text{Vap.2}}$$

```

FUNCTION EQ1
! radiation effect
REAL LOAD
CALL GETVAR ('Fume.2', 'T', T1)
CALL GETVAR ('vap.2', 'T', T2)
CALL GETUVA ('hex', 'LOAD', LOAD)
RES = LOAD * 1.0E09 / (T1**4 - T2**4)
END
!
FUNCTION EQ2
! Pressure drop
CALL SELECT ('vap.2')
CALL GETALL ('vap.2', 'ALL')
CALL PROPTY ('TOTMAS', T, P, ratespar(3), G1)
CALL PROPTY ('VG', T, P, ratespar(3), VG)
CALL GETVAR ('vap.2', 'P', P1)
CALL GETVAR ('vap.3', 'P', P2)
RES = G1 * P1 * VG / (P1*P1 - P2*P2)
END

```

The list of measurements and results given in table below.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
FLX-01_VAR1	UNIT	FLX-01	VAR1	2.4000	-5.000	-
FLX-01_VAR2	UNIT	FLX-01	VAR2	0.70000	-5.000	-
FUME.1_MASSF	STREAM	FUME.1	MASSF	3.7000	-10.00	kg/s
FUME.1_MFCO2	STREAM	FUME.1	MFCO2	8.5900	-5.000	%
FUME.1_MFN2	STREAM	FUME.1	MFN2	72.500	-5.000	%
FUME.1_MFO2	STREAM	FUME.1	MFO2	2.5000	0.500E-01	%
FUME.1_MFSO2	STREAM	FUME.1	MFSO2	0.0000	CST	%
FUME.1_P	STREAM	FUME.1	P	1.0000	-2.000	bar
FUME.1_T	STREAM	FUME.1	T	1700.0	20.0	C
FUME.2_T	STREAM	FUME.2	T	908.00	10.0	C
HEX_DP1	UNIT	HEX	DP1	0.0000	CST	bar
TUR_EFFIC	UNIT	TUR	EFFIC	0.91000	-5.000	-
VAP.1_MASSF	STREAM	VAP.1	MASSF	8.0000	-3.000	kg/s
VAP.1_P	STREAM	VAP.1	P	18.000	-2.000	bar
VAP.1_T	STREAM	VAP.1	T	225.00	2.00	C
VAP.2_T	STREAM	VAP.2	T	450.00	2.00	C
VAP.3_P	STREAM	VAP.3	P	5.0000	-2.000	bar
WTURB_POWER	STREAM	WTURB	POWER	2600.0	-0.5000	kW

TABLE 19 - MEASUREMENTS LIST FOR FLXVAL EXAMPLE

Number of tags	18					
Number of equations	20					
Number of unmeasured variables	17					
Number of measured variables	16					
Number of variables considered constant	3					
Obvious number of redundancies	3					
Total number of redundancies	3					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	0.17214					
Chi-square test value	7.81000					
Sum of squares of residues	0.353571E-008					
Number of active bounds	0					
Number of variables close to bounds	1					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
FLX-01_VAR1	2.4000	5.00 %	2.4251	3.42 %	0.04	-
FLX-01_VAR2	0.70000	5.00 %	0.70142	4.86 %	0.00	-
FUME.1_MASSF	3.7000	10.0 %	3.6977	4.15 %	0.00	kg/s
FUME.1_MFCO2	8.5900	5.00 %	8.5901	5.00 %	0.00	%
FUME.1_MFN2	72.500	5.00 %	72.506	4.88 %	0.00	%
FUME.1_MFO2	2.5000	0.500E-01	2.5000	0.500E-01	0.00	%
FUME.1_MFSO2	0.0000	CST	0.0000			%
FUME.1_P	1.0000	2.00 %	1.0000	2.00 %	0.00	bar
FUME.1_T	1700.0	20.0	1700.0	19.4	0.00	C
FUME.2_T	908.00	10.0	909.67	8.09	0.03	C
HEX_DP1	0.0000	CST	0.0000			bar
TUR_EFFIC	0.91000	5.00 %	0.90324	3.46 %	0.02	-
VAP.1_MASSF	8.0000	3.00 %	7.9366	2.10 %	0.07	kg/s
VAP.1_P	18.000	2.00 %	17.994	2.00 %	0.00	bar
VAP.1_T	225.00	2.00	225.09	1.97	0.00	C
VAP.2_T	450.00	2.00	449.89	1.98	0.00	C
VAP.3_P	5.0000	2.00 %	5.0037	1.97 %	0.00	bar
WTURB_POWER	2600.0	0.500 %	2600.2	0.499 %	0.00	kW

TABLE 20 - RESULTS OF FLXVAL EXAMPLE

SECTION 21: FLEXLIB: USER DEFINED FUNCTIONS

PURPOSE

The **FLEXLIB** unit has been designed to contain Flex user-defined functions which can be referred to in any Flex code. The functions are available throughout the complete model, even if you run only selected PFDs that do not include that **FLEXLIB** unit or if the **FLEXLIB** unit is included in a PFD that is OFF.

SYMBOLS



FIGURE 31 - SYMBOLS FOR UNIT FLEXLIB

CONNECTION RULES

No stream can be connected to a unit of type **FLEXLIB**.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	FLEXLIB used / not used

When set to "**FLEXLIB** not used", the functions defined in the unit are ignored. This enables to keep different versions of user-defined functions in a single model.

SECTION 22: GROSSERR: GROSS ERRORS ELIMINATION

PURPOSE

The **GROSSERR** unit is used to specify the settings of the Gross Error Management system, which deals with suspect measurements that may bias the results of the data reconciliation problem. The Gross Error Management system includes:

- the detection of gross errors based on quality criteria
- the identification or selection of "suspect" measurements
- the correction to be applied for the selected "suspect" measurements

Three approaches are proposed for the selection of "suspect" measurements:

- the "Highest impact" method evaluates successively the impact on the total penalty of removing each measurement. The measurement which as the highest impact is then selected and the procedure iterates as the quality criteria is satisfied. This method is recommended
- the "Highest measured penalty" method simply selects the measurements according to their penalty, calculated on basis of the measurement accuracy
- the "Highest correction penalty" method selects the measurements according to their penalty, calculated on basis of the correction accuracy (VDI recommendation)

For all approaches, the "Failure probability factor" is a (multiplying) weighting factor that is used to favor or un-favor the selection of measurements. Setting the Failure probability of a measurement to 0 will simply prevent the selection of the measurement in any case.

The criteria of quality can be either:

- that the Chi2 test is satisfied
- that the Chi2 test is satisfied and no measurement are flagged

SYMBOLS



FIGURE 32 - SYMBOLS FOR UNIT GROSSERR

CONNECTION RULES

No connections are allowed on a **GROSSERR** unit.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Mode
CRITERIA	General	Input	Detection Criterion
PENALTY	General	Input	Individual Criterion
TYPEELIM	General	Input	Type of selection
NOCONV	General	Input	Behavior when no convergence
MODE2SL	General	Input	Mode for solver switch
MODEREM E	General	Input	Mode for gross error correction
MODEBND	General	Input	Bounds on variables
FAILPROB	Parameters	Input	Failure Probability
MAXELIM	Parameters	Input	Maximum Number of iterations
MAXFAIL	Parameters	Input	Maximum failures for gross error correction
EVOCRIT	Parameters	Input	Threshold for individual criterion
EQUICRIT	Parameters	Input	Equivalent gross errors criteria
CORCRIT	Parameters	Input	Correlation criteria

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Gross Errors are not eliminated	the Gross Error management procedure is not activated
1	Gross Errors are eliminated	the Gross Error management procedure is activated

CRITERIA (General)

Detection Criterion (type: Input).

1	Chi-Square Test	Gross errors are eliminated until the Chi2 test is satisfied
2	Chi-Square & Individual Test	Gross errors are eliminated until the Chi2 test is satisfied and no individual measurement is flagged any more

PENALTY (General)

Individual Criterion (type: Input).

1	Based on measured penalty	The penalty calculated on each measurement is based on the accuracy of the measurement
2	Based on impact	The penalty calculated on each measurement is based on its impact on the weighted sum of penalty
3	Based on correction penalty	The penalty calculated on each measurement is based on the accuracy of the correction applied to the measurement (according to VDI recommendation)

TYPEELIM (General)

Type of selection (type: Input).

1	Highest measured penalty	Highest penalty: the measurements with the highest penalty (weighted by the failure probability) are removed first
2	Highest impact	Highest impact: the measurements with the highest impact on the total penalty (weighted by the failure probability) are removed first
3	Highest correction penalty	Highest correction penalty: the measurements with the highest correction penalty (weighted by the failure probability) are removed first

NOCONV (General)

Behavior when no convergence (type: Input).

0	No Elimination when no convergence	No correction when no convergence
1	Elimination allowed when near convergence	Correction allowed when near convergence

MODE2SL (General)

Mode for solver switch (type: Input).

0	Allow switch between solvers when -2SL	Allow switch between solvers (when -2SL runtime option selected)
1	No switch between solvers	No switch between solvers during gross errors management

MODEREME (Parameters)

Mode for gross error correction (type: Input).

0	Serial elimination	Serial elimination
1	Serial Relaxation of equivalencies	Serial relaxation of uncertainties of equivalent gross errors (combined with serial elimination of gross errors)
2	Serial Relaxation of all Errors	Serial relaxation of uncertainties of all gross errors

MODEBND (General)

Mode for solver switch (type: Input).

0	Solver Bounds	Solver bounds (SL or SH) are considered for gross error selection
1	Error Bounds	Error bounds (LL or HH) are considered for gross error selection

FAILPROB (Parameters)

Failure Probability (type: Input).

FAILPROB defines the default probability of tags defined from tag files. This parameter is not used if the tags are directly defined in the ValiModeller.

MAXELIM (Parameters)

Maximum Number of iterations (type: Input).

MAXELIM defines the maximum number of iterations of the gross error algorithm. A value of 0 is equivalent to set the default value (10).

MAXFAIL (Parameters)

Maximum failures for gross error correction (type: Input).

MAXFAIL defines the maximum number of convergence failure that can happen during correction of a tag by the gross error algorithm. A value of 0 is equivalent to set the default value (10).

EVOCRIT (Parameters)

Threshold for individual criterion (type: Input).

Default value = 6.6349

EVOCRIT is the threshold value used for the individual criterion. The default value corresponds to the penalty created by a single measurement when it gets out of its 99% confidence interval.

EQUICRIT (Parameters)

Equivalent gross errors criteria (type: Input).

Default value = 0.05

EQUICRIT is the threshold value used for the identification of equivalent gross errors. Gross errors are considered as equivalent if their impacts differ by less than EQUICRIT.

CORCROT (Parameters)

Correlation criteria (type: Input).

Default value = 0.8

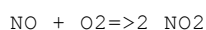
CORCRIT is the threshold value used for the identification of correlation between equivalent gross errors. Equivalent gross errors are considered as correlated if their impact variation is greater than CORCRIT

SECTION 23: HEX-HNO3: NITROUS GAS COOLER

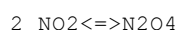
PURPOSE

This UNIT models the cooling a nitrous gas mixture, taking into account reactions in the gas phase:

peroxidation:



dimerization:



The extents of these reactions are calculated on basis of BODENSTEIN kinetic equations.

The reactions are part of the model and must not be defined in the reactions page of the unit.

SYMBOLS

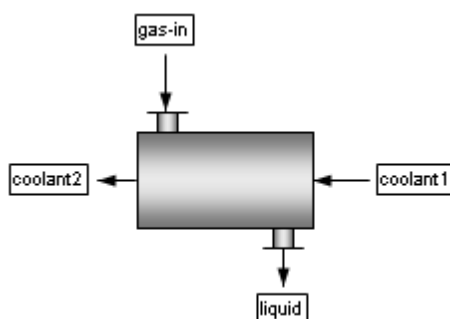


FIGURE 33 - SYMBOL FOR UNIT HEX-HNO3

CONNECTION RULES

INLET STREAMS (2)

- shell side, nitrous gas mixture (state V, LV, LLV or WLV)
- tube side, cooling media (any state)

OUTLET STREAMS (2)

- shell side, remaining nitrous gas mixture (any state)
- tube side, cooling media (any state)

THERMOD:

Both nitrous gas material streams must use the same THERMOD, which must contain the following compounds in any order : NO, NO2, N2O4, O2, N2, H2O, HNO3. Any other compound is considered as inert.

Both coolant streams(inlet and outlet) must use the same THERMOD.

REACTIONS:

The reactions are part of the model and must not be defined in the Reactions page.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
LOAD	Parameters	Result	Heat load of the exchanger
VOL	Parameters	Measurable	Shell volume
HA	Parameters	Measurable	Transfer coefficient
DTSIDE1,2	Parameters	Measurable	Temperature difference on each side
MODEDPi	Parameters	Input	Pressure equation
DPi	Parameters	Measurable	Pressure drop
DPXi	Parameters	Input	Nominal pressure drop
MASSFxi	Parameters	Input	Nominal flow rate
CFDPi	Parameters	Measurable	DP correction factor
EXTENTi	Parameters	Measurable	Extent of reactions

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Atomic and Energy Balance	Atomic and energy balances are taken into account
1	Atomic Balance only	Only atomic balances are taken into account

LOAD (Parameters)

Load of the exchanger (type: Result).

This parameter is the amount of heat transferred by the unit.

VOL (Parameters)

Shell volume (type: Measurable).

The shell volume is taken into account into the Bodenstein kinetics.

HA (Parameters)

Transfer coefficient (type: Measurable).

The heat transfer coefficient is defined as follows:

$$LOAD = HA \cdot LNTD$$

$$LNTD = \frac{DTSIDE1 - DTSIDE2}{\ln(DTSIDE1/DTSIDE2)}$$

where:

- DTSIDE1 and DTSIDE2 are the temperature differences on each side of the heat exchanger

DTSIDE1,2 (Parameters)

Temperature difference on each side (type: Measurable).

These parameters define the temperature difference of the streams at each side of the exchanger:

$$DTSIDE1 = Ti1 - To2$$

$$DTSIDE2 = To1 - Ti2$$

EXTENTi (Parameters)

Extent of reactions (type: Measurable).

The extents are defined (on molar basis) as follows:

- Extent: amount of O₂ consumed in peroxidation
- Extent2: amount of N₂O₄ produced by dimerization

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form: $DP = CFDP * DPX * (MASSF / MASSFX)^2$ where: ■ CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%) ■ DPX: the nominal pressure drop ■ MASSFX: the nominal flow rate ■ MASSF: the actual flow rate

A value of MODEDP has to be given for both outlet streams.

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{in} - P_{out}$$

This equation is not generated if MODEDP = 0.

where:

- P_{in} : the pressure of the inlet stream
- P_{out} : the pressure of the outlet stream

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when MODEDP = 2.

MASSF (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when MODEDP = 2.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

EXAMPLE : NITRIC ACID PROCESS

Attention: see model `nitric.bls` in the directory "`\examples\nitric`."

SECTION 24: HEXVAL: RIGOROUS HEAT EXCHANGER

PURPOSE

This type of UNIT allows you to consider heat and mass balances of heat exchangers involving two inlet and two outlet streams. The following equations can be handled:

- heat transfer linked to the difference of temperature between the streams
- pressure drop equations can also be generated
- saturation state of the outlet streams

The area and heat transfer coefficients as well as the temperature difference on both sides can be entered as pseudo-measurements. The heat transfer can be due to convection and/or radiation and can be made dependant on the flow rates.

The shell side and tubes side of the unit can be modeled separately as 2 different units (possibly in two different PFDs).

SYMBOLS

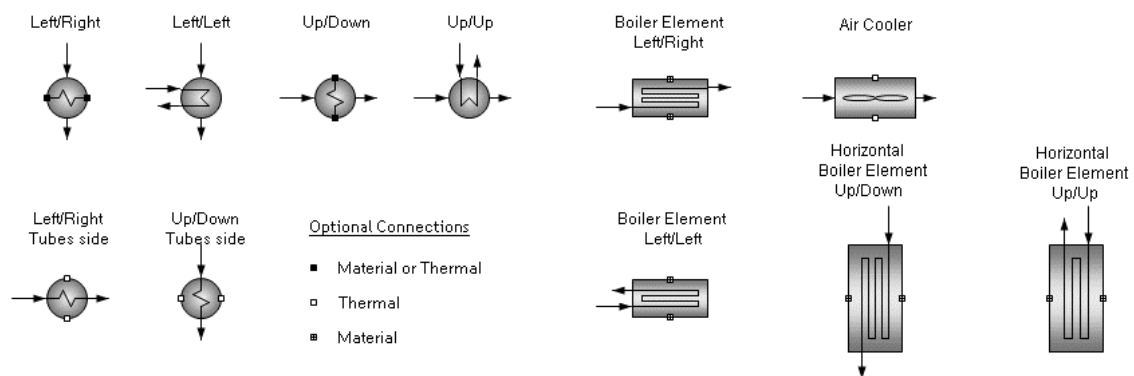


FIGURE 34 - SYMBOLS FOR UNIT HEXVAL

The symbols Left/Right and Up/Down can be used to model the complete exchanger or only its shell side. In the latter case the tubes side will be modeled via one of the two "Tubes side" symbols and the two parts of the heat exchanger will be connected by a thermal stream.

CONNECTION RULES

INLET STREAMS

- 1 or 2 material STREAMS, possibly one thermal stream

OUTLET STREAMS (2)

- 1 or 2 material STREAMS, possibly one thermal stream

Corresponding inlet and outlet STREAMS must use the same THERMOD.

The shell side is always considered as the first connections when modeling the unit in a single piece.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
CONNECT	General	Input	Connection rule
MODEU	General	Input	Mode for calculation of heat transfer coefficient
MODEPASS	General	Input	Number of passes on both sides
METHOD	General	Input	Calculation method (DTLN or NTU)
OPERMODE	General	Input	NTU Operation mode
LOAD	Parameters	Measurable	Heat load of the exchanger
AREA	Parameters	Measurable	Exchanger surface area
UGLOB	Parameters	Measurable	Heat transfer coefficient
DTSIDE1,2	Parameters	Measurable	Temperature difference on each side
RTUB	Parameters	Input	Tube resistance
CFH	Parameters	Measurable	Heat transfer correction factor
FF	Parameters	Measurable	Fouling factor
CORDTLN	Parameters	Result	DTLN correction factor
G	Parameters	Measurable	Geometric factor for radiation
DTLN	Parameters	Result	Logarithmic average of the temperature difference
MODEDPi	Parameters	Input	Pressure equation
DPi	Parameters	Measurable	Pressure drop

NAME	TAB	TYPE	DESCRIPTION
DPXi	Parameters	Input	Nominal pressure drop
MASSFxi	Parameter	Input	Nominal flow rate
CFDPi	Parameters	Measurable	DP correction factor
VFRACi	Parameters	Input	Vapor fractions of outlet streams
Ui	Parameters	Measurable	Film heat transfer coefficient
REFERUi	Parameters	Input	Nominal Ui
MASSUi	Parameters	Input	Nominal flow rate

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Mass and Energy Balance	Mass and energy balances are taken into account
1	Mass Balance only	Only the mass balances are taken into account
10	Counter-current exchanger	Counter-current performance equation is added
20	Co-current exchanger	Co-current performance equation is added
50	Radiative exchanger	Heat transfer through radiation
60	Conv + Rad Counter-current	Heat transfer through radiation and counter-current convection
70	Conv + Rad Co-current	Heat transfer through radiation and co-current convection

If the DTLN method is selected (see METHOD), the performance equation (for MODE 10 to 70) is:

$$\text{LOAD} = 10^9 \text{ VAR1} \left(T_{\text{Fume},2}^4 - T_{\text{Vap},2}^4 \right)$$

$$\text{LNTD} = \frac{\text{DTSIDE1} - \text{DTSIDE2}}{\ln(\text{DTSIDE1} / \text{DTSIDE2})}$$

where:

- DTSIDE1 and DTSIDE2: the temperature differences on each side of the heat exchanger
- σ : the constant of Boltzmann ($5.667 \cdot 10^{-8} \text{ W/m}^2\text{K}^4$)
- Thot: the outlet temperature of the hot stream
- Tcold: the outlet temperature of the cold stream
- Uglob: the heat transfer coefficient
- CORDTLN: a correction factor that accounts for the deviation from true counter-configuration (only for multi-passes exchangers, see parameter NPASSES)

If the NTU method is selected, only convective heat transfer can be modeled (MODE 10 or 20). The NTU method is based on the notion of efficiency, the efficiency being defined by the ratio between the actual heat load of the exchanger and the maximum heat load achievable assuming an infinite heat transfer surface area.

$$\text{EFF} = \text{LOAD} / \text{LOADMAX}$$

The efficiency is calculated on basis of the number of transfer unit NUT according to various formulas depending on the exchanger configuration and operation mode.

		COUNTER-CURRENT	CO-CURRENT
Single phase on both sides	$C = C_{min}/C_{max}$	$EFF = \frac{1 - e^{-NUT(1-C)}}{1 - C \cdot e^{-NUT(1-C)}}$	$EFF = \frac{1 - e^{-NUT(1+C)}}{1 + C}$
Vaporization or Condensation	$C = 0$	$EFF = 1 - e^{-NUT}$	$EFF = 1 - e^{-NUT}$

$$NUT = U_{glob} * A / C_{min}$$

$$C = C_{min} / C_{max}$$

where:

- C_{max} and C_{min} : the total heat capacities of the streams ($Massf * C_p$)

With either methods (DTLN or NTU), you can define measurements on temperature differences (DTSIDE1, DTSIDE2), heat load (LOAD), heat exchange area (A) and heat exchange coefficients (by convection U_{glob} and by radiation G). Among these three latest variables (A, U_{glob} and G), at least two of them must be, either measured, or set constant. Alternatively the heat transfer coefficient U_{glob} can be calculated in function of the flow rates of the streams via the parameter MODEU.

CONNECT (General)

Connection rule (type: Input).

0	Automatic detection	the hot and cold stream positions will be automatically detected on basis of the stream temperature data
1	Cold in Shell	the cold stream is assumed to be in the Shell side
2	Cold in Tubes	the cold stream is assumed to be in the Tubes side

MODEU (General)

Mode for calculation of heat transfer coefficient (type: Input).

0	Only Uglob is variable	No other equation is generated
1	Ushell and Utubes variables	Uglob will be calculated in function of the film heat transfer coefficients according to: $1/U_{glob} = FF + 1/(CFH \cdot U_{clean})$ where: ■ FF: a fouling factor ■ CFH: a tuning parameter ■ Uclean: the clean heat transfer coefficient calculated from: $1/U_{clean} = (1/U_1 + RTUB + 1/U_2)$ where: ■ U1: the film heat transfer coefficient on shell side ■ RTUB: the tubes thermal resistance ■ U2: the film heat transfer coefficient on tubes side
2	U=F(MASSF)	In addition to the above equations, the film heat transfer coefficients are related to the flow rates: $U_i = REF U_i * (MASS F_i / MASS F_{U_i})^{0.7}$ where: ■ REF U_i: the film heat transfer coefficient at nominal conditions ■ MASS F_i: the actual flow rate ■ MASS F_{U_i}: the flow rate at nominal conditions

MODEPASS (General)

Number of passes on both sides (type: Input).

0	1/1 or 2/2 passes	perfect counter-current configuration. The parameter CORDTLN will be equal to 1
1	1/(2 or more)	one pass on shell side, 2 or more passes on tubes side
2	2/(4 or more)	two passes on shell side, 4 or more passes on tubes side

LOAD (Parameters)

Load of the exchanger (type: Measurable).

This parameter is the amount of heat transferred by the unit.

AREA (Parameters)

Exchanger surface area (type: Measurable).

This parameter is the area of the exchanger used for heat transfer.

UGLOB (Parameters)

Heat transfer coefficient (type: Measurable).

This parameter is the heat transfer coefficient of the exchanger. It is only relevant when heat transfer through convection is taken into account.

Attention: When a convection equation is used (MODE=10, 20, 60 or 70) it is necessary to define at least one pseudo-measurement on AREA or on UGLOB.

If the heat exchanger area is unknown, you can set the surface area to 1 (Measurement with expected deviation set to zero), and use the UGLOB parameter as a global UA factor.

DTSIDE1,2 (Parameters)

Temperature difference on each side (type: Measurable).

These parameters define the temperature difference of the streams at each side of the exchanger. Streams used for definition depend on the type of exchanger.

- $DTSIDE1 = Ti1 - Ti2$ and $DTSIDE2 = To1 - To2$ for co-current arrangement
- $DTSIDE1 = Ti1 - To2$ and $DTSIDE2 = To1 - Ti2$ for counter-current arrangement

Attention: The definition of the difference of temperatures leads to positive values only if the connection is set appropriately through the parameter CONNECT. This is also the case for the heat load parameter.

RTUB (Parameters)

Tube resistance (type: Input).

CFH (Parameters)

Heat transfer correction factor (type: Measurable).

FF (Parameters)

Fouling factor (type: Measurable).

CORDTLN (Parameters)

DTLN correction factor (type: Result).

These parameters are used in the performance equations describe above.

G (Parameters)

Geometric factor for radiation (type: Measurable).

This factor is only relevant when radiation is taken into account (Mode = 50, 60 or 70). It includes the effect of emissivity as well as the geometric factor.

Attention: When the radiation equation is used (MODE=50, 60 or 70) it is necessary to define at least one pseudo-measurement on A or on G.

If the heat exchanger area is unknown, you can set the surface area to 1 (Measurement with expected deviation set to zero), and use the G parameter as a global GA factor.

DTLN (Parameters)

Logarithmic average of the temperature difference (type: Result).

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * (MASSF / MASSFX)^2$$

where:

- CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%)
- DPX: the nominal pressure drop
- MASSFX: the nominal flow rate
- MASSF: the actual flow rate

A value of MODEDP_i has to be given for each outlet stream, although it is only meaningful for material streams.

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{in} - P_{out}$$

This equation is not generated if MODEDP = 0.

where:

- P_{in} : the pressure of the inlet stream
- P_{out} : the pressure of the outlet stream

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when MODEDP = 2.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when MODEDP = 2.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when MODEDP = 2.

VFRAC1,2 (Parameters)

Saturation state (type: Input).

A value of VFRAC must be given for each outlet STREAM, although it will only be meaningful for material STREAMS:

-1	-	no saturation constraint for the corresponding outlet stream
0	-	the corresponding stream is at bubble point. It must be defined as Liquid
1	-	the corresponding stream is at dew point. It must be defined as Vapor

These parameters define the vapor ratio in the outlet stream. The value of this parameter has to be set in the range of 0 to 1 : if the value is out of this range, saturation constraint is not considered. This parameter is always a constant for the model.

Attention:

1. A Value of 0 corresponds to a saturated liquid (bubble point); a value of 1 corresponds to a saturated vapor (dew point).
2. If you want to fix bubble or dew point with VFRAC, it is highly recommended to define the corresponding Streams with L or V type instead of LV. This avoids mathematical instabilities due to proximity of saturation.

Ui (Parameters)

Film heat transfer coefficient (type: Measurable).

REFERUi (Parameters)

Nominal Ui (type: Input).

MASSUi (Parameters)

Nominal flow rate (type: Input).

Parameters used to calculate the film heat transfer coefficients in function of flow rates

EXAMPLE : EXCHANGER WITH STEAM VAPORIZATION

Attention: see model `hexval.bls` in the directory "examples\hexval".

This example is a heat exchange between fumes and steam in a vaporizer.

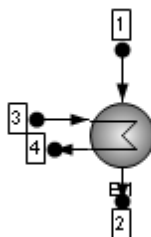


FIGURE 35 - AN EXAMPLE OF HEXVAL TYPE UNIT

There is 1 UNIT:

- E-1:
 - type=HEXVAL
 - SYMBOL='Exchanger Left/Left'
 - MODE='Counter-current exchanger'
 - CONNECT='Cold in Tubes'
 - MODEU='U=F(MASSF)'
 - MODEPASS='1/1 or 2/2 passes'
 - VFRAC1=1 (stream HOT2 at dew point)
 - MODEDP1 = 'DP=F(MASSF)', DPX1 = 2 bar, MASSFX1 = 500 t/h
 - MODEDP2 = 'DP=F(MASSF)', DPX2 = 1 bar, MASSFX2 = 80 t/h

4 STREAMS must be defined:

STREAMS	THERMOD	CONNECTIONS
1	FUMES	to E-1
2	FUMES	from E-1
3	STEAM	to E-1
4	STEAM	from E-1

The THERMODS STEAM and FUMES are defined in the file Empty.bls.

The list of measurements and the results are given below.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
1_MASSF	STREAM	1	MASSF	500.00	-10.00	t/h
1_MFCO2	STREAM	1	MFCO2	8.5900	-5.000	%
1_MFH2O	STREAM	1	MFH2O	15.000	-5.000	%
1_MFN2	STREAM	1	MFN2	72.500	-5.000	%
1_MFO2	STREAM	1	MFO2	2.5000	0.500E-01	%
1_MFSO2	STREAM	1	MFSO2	0.0000	CST	%
1_P	STREAM	1	P	200.00	-2.000	bar
1_T	STREAM	1	T	550.00	5.00	C
2_P	STREAM	2	P	198.00	-2.000	bar
2_T	STREAM	2	T	400.00	5.00	C
3_MASSF	STREAM	3	MASSF	65.000	-5.000	t/h
3_P	STREAM	3	P	85.000	-2.000	bar
3_T	STREAM	3	T	298.00	1.00	C
4_P	STREAM	4	P	84.000	-2.000	bar
4_T	STREAM	4	T		OFF	C
E-1_AREA	UNIT	E-1	AREA	500.00	CST	m2
E-1_CFDP1	UNIT	E-1	CFDP1	1.0000	-30.00	-
E-1_CFDP2	UNIT	E-1	CFDP2	1.0000	-30.00	-
E-1_CFH	UNIT	E-1	CFH	1.0000	CST	-
E-1_FF	UNIT	E-1	FF	0.0000	10.0	m2*s*K/kJ

TABLE 21 - MEASUREMENT LIST FOR HEXVAL EXAMPLE

Number of tags	20
Number of equations	24
Number of unmeasured variables	19
Number of measured variables	16
Number of variables considered constant	4
Obvious number of redundancies	5
Total number of redundancies	5
Number of trivial redundancies	0

Validation and reconciliation results

Objective function	0.16508
Chi-square test value	11.0800
Sum of squares of residues	0.144777E-009
Number of active bounds	0
Number of variables close to bounds	0
Number of eliminated tags	0
Number of flagged tags	0
Number of filtered tags	0

TAGS IN THE MODEL

Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
1_MASSF	500.00	10.0 %	499.06	5.68 %	0.00	t/h
1_MFCO2	8.5900	5.00 %	8.6087	4.96 %	0.00	%
1_MFH2O	15.000	5.00 %	15.056	4.87 %	0.01	%
1_MFN2	72.500	5.00 %	73.835	1.14 %	0.14	%
1_MFO2	2.5000	0.500E-01	2.5003	0.500E-01	0.00	%
1_MFSO2	0.0000	CST	0.0000			%
1_P	200.00	2.00 %	200.00	1.42 %	0.00	bar
1_T	550.00	5.00	549.97	4.80	0.00	C
2_P	198.00	2.00 %	198.00	1.43 %	0.00	bar
2_T	400.00	5.00	400.03	4.81	0.00	C
3_MASSF	65.000	5.00 %	65.044	4.55 %	0.00	t/h
3_P	85.000	2.00 %	84.829	1.41 %	0.01	bar
3_T	298.00	1.00	298.00	0.999	0.00	C
4_P	84.000	2.00 %	84.166	1.42 %	0.01	bar
4_T		OFF	298.57	1.02		C
E-1_AREA	500.00	CST	500.00			m2
E-1_CFD1	1.0000	30.0 %	1.0000	29.8 %	0.00	-
E-1_CFD2	1.0000	30.0 %	1.0035	29.8 %	0.00	-
E-1_CFH	1.0000	CST	1.0000			-
E-1_FF	0.0000	10.0	0.36864	0.148	0.00	m2*s*K/kJ

TABLE 22 - RESULTS OF HEXVAL EXAMPLE

SECTION 25: LLEVAL: LIQUID-LIQUID-SEPARATOR

PURPOSE

The **LLEVAL** unit handles mass balances, energy balance and liquid-liquid equilibrium relationships of a liquid-liquid separator. Heat can be extracted or provided to the separator via Thermal streams.

The temperature and the pressure of the liquid outlets are equal and it is possible either to consider a pressure drop in the separator or to consider isobaric behavior.

SYMBOLS

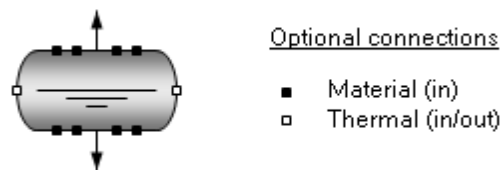


FIGURE 36 - SYMBOLS FOR UNIT LLEVAL

CONNECTION RULES

INLET STREAMS

- Maximum number:
 - Material: 8
 - Thermal: 2

OUTLET STREAMS

- Maximum number:
 - Material: 2 streams of state L, LL or WL
 - Thermal: 2

THERMODS

Both outlet streams must use the same THERMOD.

Attention:

1. The state of the outlet material streams can also be LV, LLV and WLV provided that there is more than one component in the stream.

- It is essential to initialize the composition of the outlet streams before attempting to activate the L/L equations. Failure to do so can induce convergence problems or lead to a trivial solution where both outlet streams have the same composition. The initialization can be performed manually or automatically via the parameter FLASH.
- No assumption is made about the destination of solids that would be present in the inlet streams. Composition measurements (or information coming from the downstream balances) should define the split of solids between the two liquid phases.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
FLASH	General	Input	Flash initialization
REFDP	Parameters	Input	Reference stream for pressure drop
DP	Parameters	Measurable	Pressure drop

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	All equations	Mass, energy balances and liquid-liquid equilibrium relationships are taken into account
1	Mass Balance only	Only the mass balances are taken into account
2	Mass and Energy Balance	Mass and energy balances are taken into account
3	Mass and L/L equilibrium	Mass and liquid-liquid equilibrium relationships are taken into account

Attention:

- The outlet streams are always set at the same pressure and temperature except in Mode 1 (Mass balance only).
- If the partial flow rate of some of the compounds is detected as being equal to zero in one of the Streams, the related equilibrium relationship is not considered.

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

FLASH (General)

Flash initialization (type: Input).

0	None	no initialization of outlet streams (keep values of last run)
1	Always	initialization in all cases on basis of input streams data
2	When needed	initialization on basis of input streams data in the following cases: ▪ one of the flow rates of the two first outlet streams is equal to 0 ▪ the temperature of one of the two first outlet streams is equal to 0°C

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

FLASH (General)

Flash initialization (type: Input).

0	None	no initialization of outlet streams (keep values of last run)
1	Always	initialization in all cases on basis of input streams data
2	When needed	initialization on basis of input streams data in the following cases: ■ one of the flow rates of the two first outlet streams is equal to 0 ■ the temperature of one of the two first outlet streams is equal to 0°C

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

SECTION 26: LLVVAL : LIQUID-LIQUID-VAPOR SEPARATOR

PURPOSE

This type of UNIT handles mass balances, energy balance and liquid-liquid-vapor equilibrium relationships of a 3 phase separator. Heat can be extracted or provided to the separator via Thermal streams.

The temperature and the pressure of the liquid and vapor outlets are equal and it is possible either to consider a pressure drop in the separator or to consider isobaric behavior.

SYMBOLS

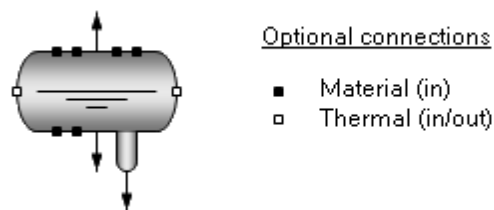


FIGURE 37 - SYMBOLS FOR UNIT LLVVAL

CONNECTION RULES

INLET STREAMS

- Maximum number:
 - Material: 6
 - Thermal: 2

OUTLET STREAMS

- Maximum number:
 - Material:
 - 2 streams of state L, LL or WL for the liquids
 - 1 stream of state V
 - Thermal: 2

THERMODS

All outlet streams must use the same THERMOD.

Attention:

1. The state of the outlet material streams can also be LV, LLV and WLV provided that there is more than one component in the stream.
2. It is essential to initialize the composition of the outlet streams before attempting to activate the L/L/V equations. Failure to do so can induce convergence problems or lead to a trivial solution where all outlet streams have the same compositions. The initialization can be performed manually or automatically via the parameter FLASH.
3. No assumption is made about the destination of solids that would be present in the inlet streams. Composition measurements (or information coming from the downstream balances) should define whether solids are going with the liquid or vapor phases.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
FLASH	General	Input	Flash initialization
REFDP	Parameters	Input	Reference stream for pressure drop
DP	Parameters	Measurable	Pressure drop

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	All equations	Mass, energy balances and liquid-liquid-vapor equilibrium relationships are taken into account
1	Mass Balance only	Only the mass balances are taken into account
2	Mass and Energy Balance	Mass and energy balances are taken into account
3	Mass and L/L/V equilibrium	Mass and liquid-liquid-vapor equilibrium relationships are taken into account

Attention:

1. The outlet streams are always set at the same pressure and temperature except in Mode 1 (Mass balance only).

2. If the partial flow rate of some of the compounds is detected as being equal to zero in one of the Streams, the related equilibrium relationship is not considered.

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

FLASH (General)

Flash initialization (type: Input).

0	None	no initialization of outlet streams (keep values of last run)
1	Always	initialization in all cases on basis of input streams data
2	When needed	initialization on basis of input streams data in the following cases: ▪ one of the flow rates of the two first outlet streams is equal to 0 ▪ the temperature of one of the two first outlet streams is equal to 0°C

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the reference inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if $MODEDP = 0$.

SECTION 27: LOFUNIT (PFD): LIST OF UNITS

PURPOSE

The purpose of a **LOFUNIT** (or PFD) is to define a List OF UNITS which are modeled together.

A maximum of 99 units are allowed in a **LOFUNIT**. A **LOFUNIT** may itself contain other units of type **LOFUNIT**, up to a maximum level of embedding of 9.

SYMBOLS

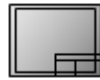


FIGURE 38 - SYMBOLS FOR UNIT LOFUNIT

CONNECTION RULES

No connections are allowed on a **LOFUNIT**.

UNIT PARAMETERS (SUMMARY)

This unit has no parameter.

SECTION 28: LVEVAL : LIQUID-VAPOR SEPARATOR

PURPOSE

The LVEVAL unit handles mass balances, energy balance and liquid-vapor equilibrium relationships of a liquid-vapor separator. Heat can be extracted or provided to the separator via Thermal Streams.

The temperature and the pressure of the outlets are equal and it is possible either to consider a pressure drop in the separator or to consider isobaric behavior.

SYMBOLS

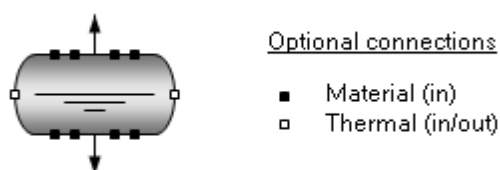


FIGURE 39 - SYMBOLS FOR UNIT LVEVAL

CONNECTION RULES

INLET STREAMS

- Maximum number:
 - Material: 8
 - Thermal: 2

OUTLET STREAMS

- Maximum number:
 - Material:
 - 1 stream of state V as vapor outlet (upper stream)
 - 1 stream of state L as liquid outlet (lower stream)
 - Thermal: 2

THERMODS

Both outlet streams must use the same THERMOD.

Attention:

1. The state of the outlet material streams can also be LV, LLV and WLV provided that there is more than one component in the stream.

2. It is essential to initialize the composition of the outlet streams before attempting to activate the L/V equations. Failure to do so can induce convergence problems or lead to a trivial solution where both outlet streams have the same composition. The initialization can be performed manually or automatically via the parameter FLASH.
3. No assumption is made about the destination of solids that would be present in the inlet streams. Composition measurements (or information coming from the downstream balances) should define whether solids are going with the liquid or vapor phase.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
FLASH	General	Input	Flash initialization
REFDP	Parameters	Input	Reference stream for pressure drop
DP	Parameters	Measurable	Pressure drop

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	All equations	Mass, energy balances and liquid-vapor equilibrium relationships are taken into account
1	Mass Balance only	Only the mass balances are taken into account
2	Mass and Energy Balance	Mass and energy balances are taken into account
3	Mass and L/V equilibrium	Mass and liquid-vapor equilibrium relationships are taken into account

Attention:

1. The outlet streams are always set at the same pressure and temperature except in Mode 1 (Mass balance only).
2. If the partial flow rate of some of the compounds is detected as being equal to zero in one of the Streams, the related equilibrium relationship is not considered.

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

FLASH (General)

Flash initialization (type: Input).

0	None	no initialization of outlet streams (keep values of last run)
1	Always	initialization in all cases on basis of input streams data
2	When needed	initialization on basis of input streams data in the following cases: ▪ one of the flow rates of the two first outlet streams is equal to 0 ▪ the temperature of one of the two first outlet streams is equal to 0°C

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

EXAMPLE : FLASH OF A MIXTURE OF O2, N2, AR

Attention: see model lveval.bls in the directory "examples\lveval".

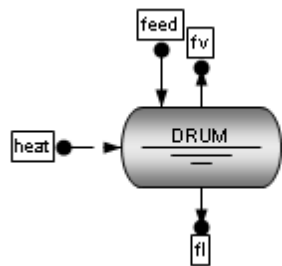


FIGURE 40 - EXAMPLE OF LVEVAL TYPE UNIT

A stream made of O2, N2 and AR is cooled down and flashed in a drum.

There is 1 UNIT :

- DRUM:
 - type=LVEVAL
 - MODE='All Equations'
 - MODEDP='DP variable'

4 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
FEED	STEAM	to DRUM
HEAT	THERMAL STREAM	to DRUM
FV	STEAM (Vapor)	from DRUM
FL	STEAM (Liquid)	from DRUM

The THERMOD is defined in the model file Empty.bls.

The list of measurements and the results are given in tables below.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
DRUM_DP	UNIT	DRUM	DP	1.0000	0.100	bar
FEED_MASSF	STREAM	FEED	MASSF	30.000	-5.000	t/h
FEED_P	STREAM	FEED	P	11.000	-2.000	bar
FEED_T	STREAM	FEED	T	150.00	1.00	C
FL_MASSF	STREAM	FL	MASSF	100.00	-5.000	kg/h
FV_MASSF	STREAM	FV	MASSF	29.000	-5.000	t/h
FV_P	STREAM	FV	P	10.000	-2.000	bar
HEAT_LOAD	STREAM	HEAT	LOAD	17000.	-10.00	kW

TABLE 23 - MEASUREMENT LIST FOR LVEVAL EXAMPLE

Number of tags	8					
Number of equations	12					
Number of unmeasured variables	9					
Number of measured variables	8					
Number of variables considered constant	0					
Obvious number of redundancies	3					
Total number of redundancies	3					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	0.27203					
Chi-square test value	7.81000					
Sum of squares of residues	0.144605E-019					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
DRUM_DP	1.0000	0.100	1.0000	0.948E-01	0.00	bar
FEED_MASSF	30.000	5.00 %	29.430	3.33 %	0.14	t/h
FEED_P	11.000	2.00 %	11.000	1.43 %	0.00	bar
FEED_T	150.00	1.00	150.01	1.00	0.00	C
FL_MASSF	100.00	5.00 %	100.01	5.00 %	0.00	kg/h
FV_MASSF	29.000	5.00 %	29.330	3.34 %	0.05	t/h
FV_P	10.000	2.00 %	9.9999	1.54 %	0.00	bar
HEAT_LOAD	17000.	10.0 %	17468.	3.34 %	0.08	kW

TABLE 24 - RESULTS OF LVEVAL EXAMPLE

SECTION 29: NETVAL : PIPES NETWORK FOR MASS BALANCE

PURPOSE

This unit type has been primarily developed to model the complex network of pipes connecting process units and a tank farm where the various tanks can be used on various products.

All units of type **NETVAL** (in a given model) behave like a single Black Box where all streams coming in or going out participate to different mass balances depending on a product attribute. This product type is defined as a **PRODUCT** attribute via a MEA file. It can also be set by the first 8 characters of the stream comments. A **PRODUCT** attribute can also be set on a Tank (TNKVAL unit) in which case both inlet and outlet streams will receive the corresponding attribute.

The **BALANCE PRODUCT** syntax in a mea file automatically generates tags for each specified products giving the opening and closing inventory, the charge and rundown from tanks, the import and export as well as the production and consumption per product.

In addition, the **IMBALANCE PRODUCT** syntax enables an imbalance on the selected product and generates an additional tag giving the identified product imbalance.

SYMBOLS

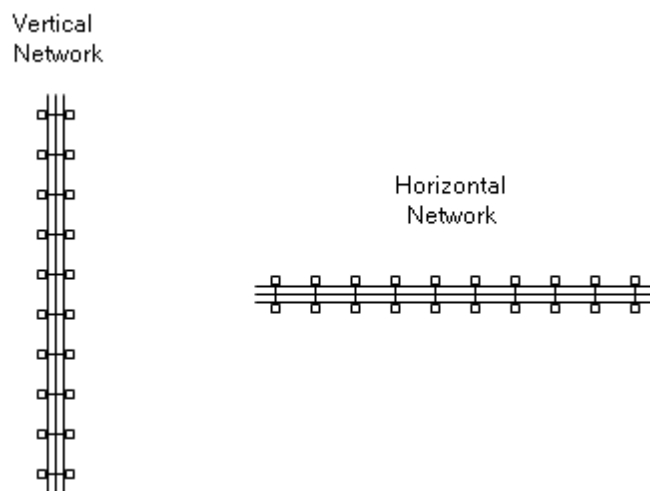


FIGURE 41 - SYMBOLS FOR UNIT NETVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 10
- Type: material

OUTLET STREAMS

- Maximum number: 10
- Type: material

Each connected stream must have an attribute.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
EQUAMODE	General	Input	Outlet Compositions Equalities

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

1 Mass Balance only Only the mass balance is considered.

EQUAMODE (General)

Outlet Compositions Equalities (type: Input).

0 None The compositions of outlet streams are free

1 Outlet All outlet streams having the same attribute will have the same composition

 Compositio

 ns

 Equalities

EXAMPLE : GAS RECOVERY UNIT

Attention: see model netval.bls in the directory "examples\netval"

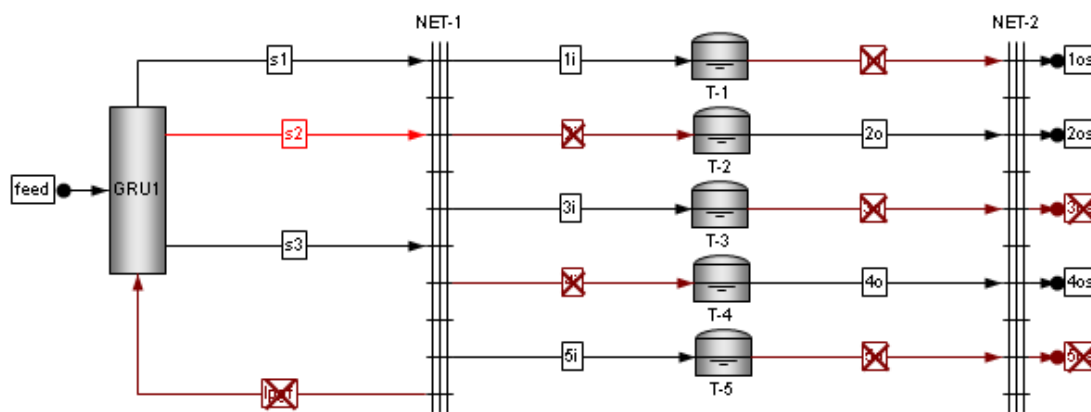


FIGURE 42 - AN EXAMPLE WITH NETVAL AND TNKVAL UNITS

This example shows the use of Netval units (NET-1 and NET-2) to interconnect a process unit (GRU1, Gas Recovery Unit) to five storage tanks (T-1 to T-5).

There are 8 UNITS:

- MIX
 - type=BBXVAL,
 - MODE='Mass Balance Only'
 - MODEDP='None'
- NET-1, NET-2
 - type=NETVAL
 - Outlet Compositions Equalities='None'
- T-1 to T-5
 - type=TNKVAL,
 - MODE='No simultaneous Input and Output'

There are 20 Streams connecting the units as indicated in the flow-sheet. They all use the same Thermod Steam. (In fact, any thermod involving a single compound could be used as only global mass balances are to be performed).

The comments of the streams and the list of measurements and the results are given in tables below.

Attention:

1. It is assumed that tags have been defined for Mass Flow Rate on each process (feed, s1, s2, etc.) and export stream (1os, 2os, ...) and on the initial and final levels of each tank.
2. The comments and measurements must be given as a measurement file.

3. The CUTOFF unit automatically sets OFF the streams for which the measured flow rate is below 0.01 t/h.
4. Stream LPGF (LPG failed) is automatically switched off because no tank contains failed LPG.

```

COMMENT STREAM 1I "C3"
COMMENT STREAM 2I "C3"
COMMENT STREAM 3I "C4"
COMMENT STREAM 4I "C4"
COMMENT STREAM 5I "C5"

COMMENT STREAM 1O "C3"
COMMENT STREAM 2O "C3"
COMMENT STREAM 3O "C4"
COMMENT STREAM 4O "C4"
COMMENT STREAM 5O "C5"

COMMENT STREAM 1OS "C3"
COMMENT STREAM 2OS "C3"
COMMENT STREAM 3OS "C4"
COMMENT STREAM 4OS "C4"
COMMENT STREAM 5OS "C5"

CUTOFF MASSF 0.01 t/h

FEED MASSF 100 -5 t/h
S1_MASSF 60 -5 t/h
S2_MASSF 30 -5 t/h
S3_MASSF 20 -5 t/h

1OS_MASSF 500 -0.5 t/d
2OS_MASSF 1000 -0.5 t/d
3OS_MASSF 0 -0.5 t/d
4OS_MASSF 500 -0.5 t/d
5OS_MASSF 0 -0.5 t/d

T-1_MASS1 1000 5 t
T-1_MASS2 1500 5 t
T-2_MASS1 2000 5 t
T-2_MASS2 1500 5 t
T-3_MASS1 2000 5 t
T-3_MASS2 3000 5 t
T-4_MASS1 2000 5 t
T-4_MASS2 1000 5 t
T-5_MASS1 1000 5 t
T-5_MASS2 1500 5 t

```

TABLE 25 - MEASUREMENT FILE FOR NETVAL/TNKVAL EXAMPLE

Number of tags	19
Number of equations	24
Number of unmeasured variables	20
Number of measured variables	17
Number of variables considered constant	0
Obvious number of redundancies	4
Total number of redundancies	4
Number of trivial redundancies	0

Validation and reconciliation results

Objective function	36.7116
Chi-square test value	9.49000
Sum of squares of residues	0.128678E-015
Number of active bounds	0
Number of variables close to bounds	0
Number of eliminated tags	0
Number of flagged tags	1
Number of filtered tags	0

TAGS IN THE MODEL

Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
1OS_MASSF	500.00	0.500 %	499.88	0.500 %	0.00	t/d
2OS_MASSF	1000.0	0.500 %	999.53	0.499 %	0.01	t/d
4OS_MASSF	500.00	0.500 %	500.94	0.498 %	0.14	t/d
FEED_MASSF	100.00	5.00 %	104.65	0.651 %	0.86	t/h
S1_MASSF	60.000	5.00 %	62.397	0.752 %	0.64	t/h
S2_MASSF	* 30.000	5.00 %	21.496	1.91 %	32.14	t/h
S3_MASSF	20.000	5.00 %	20.752	1.36 %	0.57	t/h
T-1_MASS1	1000.0	5.00	1000.5	4.98	0.01	t
T-1_MASS2	1500.0	5.00	1499.5	4.98	0.01	t
T-2_MASS1	2000.0	5.00	2000.5	4.98	0.01	t
T-2_MASS2	1500.0	5.00	1499.5	4.98	0.01	t
T-3_MASS1	2000.0	5.00	1996.3	4.95	0.56	t
T-3_MASS2	3000.0	5.00	3003.7	4.95	0.56	t
T-4_MASS1	2000.0	5.00	1996.3	4.95	0.56	t
T-4_MASS2	1000.0	5.00	1003.7	4.95	0.56	t
T-5_MASS1	1000.0	5.00	1001.0	4.90	0.04	t
T-5_MASS2	1500.0	5.00	1499.0	4.90	0.04	t

TABLE 26 - RESULTS OF NETVAL/TNKVAL EXAMPLE

SECTION 30: OPTVAL : VARIOUS VALI PARAMETERS

PURPOSE

The **OPTVAL** unit allows to define various mathematical and reporting settings

SYMBOLS



FIGURE 43 - SYMBOLS FOR UNIT OPTVAL

CONNECTION RULES

No connections are allowed on an **OPTVAL**.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
RESPRINT	General	Input	Trace in r3v report
INIT	General	Input	Type of initialization
NEGATFLR	General	Input	Handling of negative flow rates
EQUASOL V	General	Input	Unvalidable measurements
PRESMODE	General	Input	Defines the way stream pressures are handled
FLEXMODE	General	Input	Flex code processing
SPLMODE	General	Input	SPLVAL processing for single compound mixtures
SIGOR95C	General	Input	Accuracy definition
PENALSTD	General	Input	Penalty definition
TRIVRED	General	Input	Trivial redundancies
CSTKEEP	General	Input	Constants

NAME	TAB	TYPE	DESCRIPTION
UNCTHER	General	Input	Uncertainties on thermodynamics
ITMAX	Parameters	Input	Maximum number of iterations
EPS	Parameters	Input	Precision required on results
VALIDITY	Parameters	Input	Threshold for flagging corrected measurements
JACFREQU	Parameters	Input	Jacobian computation frequency
CONSTR	Parameters	Input	Algorithm for out of bounds variables
CHI2CRIT	Parameters	Input	Criterion for Chi square test
BNDTEST	Parameters	Input	Threshold for flagging active bounds
BNDCTRL U	Parameters	Input	Bound Control Uncertainty

UNIT PARAMETERS

RESPRINT (General)

Trace in r3v report (type: Input).

0	0: No	Values of variables and equation residues are not printed in the r3v report at each iteration
1	1: Yes	Values of variables and equation residues are printed in the r3v report at each iteration

INIT (General)

Type of initialization (type: Input).

0	0: Use data	Starting values for the state and link variables are the value available in the data. If these starting values are not coherent, arbitrary values are chosen
1	1: measured values	Use Starting values for variables and the link variables are the measured values. For the unmeasured variables, the default values are used. When the initial values are not valid, arbitrary values are chosen (example: negative flow rate)

INIT < 0 selects INIT = 0.

Attention: When the variables are initialized with the available measurements, composition and total flow rate measurements are used to initialize the partial flow rates.

NEGATFLR (General)

Handling of negative flow rates (type: Input).

0	0:accepted (Soldog only)	When a partial flow rate is not used in thermodynamic calculations, the program accepts a negative value for this variable (no thermal balance on the corresponding streams)
1	1:refused (Soldog only)	Negative values are not accepted

If NEGATFLR different from 1 or 0, it selects 1. This parameter is only relevant for the solver SOLDG.

EQUASOLV (General)

Unvalidable measurements (type: Input).

0	0:are kept variable	All measurements are taken into account
1	1:are set constant	Measured variables which cannot be validated will be considered constant and will no longer be taken into account in the validation problem

If EQUASOLV < 0 (or > 1), it selects 0.

PRESMODE (General)

Defines the way stream pressures are handled (type: Input).

0	0:DP constant not measured	Stream pressures are handled as any other variable. Pressure drops are considered constant if not measured
1	1:P and DP constant not measured	Stream pressures are considered constant when no measurement are defined on them. This was the behavior of VALI II
2	2:P and DP are variable	Stream pressures and the unit pressure drops are always considered as variables

FLEXMODE (General)

Flex code processing (type: Input).

0	Interpreted Flex code is interpreted at each execution of the code (slower)
1	Compiled Flex code is interpreted and compiled during the verification phase (faster)

Attention:

1. FLEXMODE 0 is available only to insure 100% compatibility with preceding versions. The compilation of the code is however not expected to cause problems of any order.
2. The use of user-defined routines is only possible when the code is compiled.

SPLMODE (General)

SPLVAL processing for single compound mixtures (type: Input).

0	Normal behavior	Streams coming from a SPLVAL always share the same record for their partial flow rates
1	Optimized behavior	Single compounds streams do not share their rates, which diminishes the non-linearity of the system of equations

Attention: SPLMODE 0 is available to insure 100% compatibility with preceding versions. The optimized behavior is however not expected to cause problems of any order.

SIGOR95C (General)

Accuracy definition (type: Input).

0	Standard deviation	Accuracy of measurements is based on their standard deviation
1	95% Confidence level (VDI- 2048)	Accuracy of measurements is based on their 95% confidence level (VDI mode)

PENALSTD (General)

Penalty definition (type: Input).

- 0 based on Penalty of individual measurements is calculated on basis of the measured accuracy
- 1 based on Penalty of individual measurements is calculated on basis of the accuracy of correction the correction (VDI mode) accuracy (VDI-2048)
- 2 based on Penalty of individual measurements is calculated on basis of their impact to impact the total penalty of the model, which is the decrease of the total penalty when the corresponding measurement is removed from the data set

TRIVRED (General)

Trivial redundancies (type: Input).

- 0 based on Trivial redundancies are kept into the system measured accuracy
- 1 based on Trivial redundancies are removed from the optimization problem at correction calculation time. All reports are still produced as if the trivial redundancies accuracy were kept. (VDI-2048)

Removing trivial redundancies can significantly improve the convergence properties of the model.

CSTKEEP (General)

Constants (type: Input).

- 0 are set as Constants are removed from the system. The corresponding variables are constants directly set to their appropriate values
- 1 are set via Constants (except partial rates that are set to zero) are kept as variables of an equation the system and set constant through a link equation
- 2 are set as Constants (except partial rates that are set to zero) are kept as variables of measurements the system and set a measurements with an uncertainty of 10^{-6} %
nts
(accuracy:
1E-06%)

Setting the constants via equations or via measurements with a low uncertainty can sometimes improve the convergence properties of a model.

Setting the constants as measurements enables them to be eliminated by the gross error algorithm. They will also appear in the results of the sensitivity analysis tool.

UNCTHER (General)

Uncertainties on thermodynamics (type: Input).

0	none	Uncertainties of thermodynamics are neglected
1	activated	Uncertainties of thermodynamics are taken into account, according to VDI-(VDI-2048) 2048. Adequate values are only available in VALI for IAPWS-97 methods

ITMAX (Parameters)

Maximum number of iterations (type: Input).

ITMAX =< 0 selects 100 iterations.

EPS (Parameters)

Precision required on results (type: Input).

Default value = 1.e-06

EPS =< 0 selects 1.e-06

This value is used to evaluate the value of the sum of the squares of the residues, for which convergence is considered. The default value is appropriate for molar rates in the range 0.01 to 1 kmol/s. If you work with significantly lower flow rates, you may have to reduce the value of this parameter.

VALIDITY (Parameters)

Threshold for flagging corrected measurements (type: Input).

Tags for which the validated value is outside the corresponding confidence interval will be flagged. For the default value of 95%, this corresponds to a difference (between measured and validated values) of about twice the expected deviation.

JACFREQU (Parameters)

Jacobian computation frequency (type: Input).

If JACFREQU =< 0 , it selects 5.

The Jacobian is estimated after JACFREQU iterations, as long as this value is satisfactory for the resolution routine. This parameter is only relevant when using the solver SOLDG.

CONSTR (Parameters)

Algorithm for out of bounds variables (type: Input).

This parameter defines the action on variables when one of these variables violates the constraints submitted to iteration (SOLDG solver only).

If $\text{CONSTR} \leq 0$, it selects 2.1

Fractional part = 0 selects the fractional part 1

Whole part of $\text{CONSTR} = 2$:

The array of variations to which variables should be submitted is multiplied by a scalar, so as to keep variables within an agreed scope at a distance from the constraints. This is equal to :

maximal variation * fractional part of CONSTR

Whole part of $\text{CONSTR} = 1$:

Only the variables that violate the constraints are brought back within agreed space at a distance from the constraints, which is equal to :

maximal variation * fractional part of CONSTR .

CHI2CRIT (Parameters)

Criterion for Chi square test (type: Input).

This parameter determines the level of confidence for the chi-square test. The chi-square is a statistical that gives, for the given confidence level, a threshold for the total sum of measurements errors. When the sum of penalty is above this threshold, there is suspicion that gross errors exist.

BNDTEST (Parameters)

Threshold for flagging active bounds (type: Input).

This parameter controls the generation of warning messages for active bounds. Bounds are considered active when the distance between the bound and the variable value is less than a threshold defined as $\text{BNDTEST} * \text{REC.ACC}$. By default, this threshold corresponds to the validated accuracy. Setting a higher value for BNDTEST will flag more variables as being on their bounds.

BNDCTRLU (Parameters)

Bound Control Uncertainty (type: Input).

This parameter controls the uncertainty set on the bound control auxiliary variable which is activated when a variable gets close to its bound.

SECTION 31: OPXVAL : USER DEFINED EQUATIONS

PURPOSE

This type of UNIT allows to add up to 9 simple user's equations, involving process variables.

These equations are of the form:

$$\text{VAR} = \text{X1} \# \text{X2}$$

where X1 and X2 are any Unit or Stream variable and # is one of the four basic mathematical operators (+, -, *, /). Variables X1, X2, X3, etc. are designed by a TAG.

The result VAR is stored in the parameters VAR1..9 of the unit and can be measured. These parameters are dimensionless.

SYMBOLS



FIGURE 44 - SYMBOLS FOR UNIT OPXVAL

CONNECTION RULES

No connections are allowed on an **OPXVAL**.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	Control Tags	Input	Type of operation
NEQU	Control Tags	Input	Number of equations
VARi	Parameters	Measurable	Results of the equations (i=1..9)

UNIT PARAMETERS

MODE (Control Tags)

Type of operation (type: Input).

0	<undefined	Not defined. A proper value must be set
	>	
1	+	Addition
2	-	Substraction
3	*	Multiplication
4	/	Division

NEQU (Control Tags)

Number of equations (type: Input).

This parameter represents the number of equations to be considered (ranges from 1 to 9).

In the same page you will specify the tags that are to be used for the equations. These tags do not need to be defined, provided their name does follow the syntax used for building default tag names from the stream or unit name and the parameters. For example the temperature of the stream LA11 can be defined through a tag as LA11_T.

If you want to use existing tags, you can simply drag them from the **Tags defined** table to the **Controlled variables** table. Use the **Mask** to limit the number of tags shown in the **Tags defined** table.

If you want to create a virtual tag, enter its name in the Mask and drag it to the **Controlled variables** table.

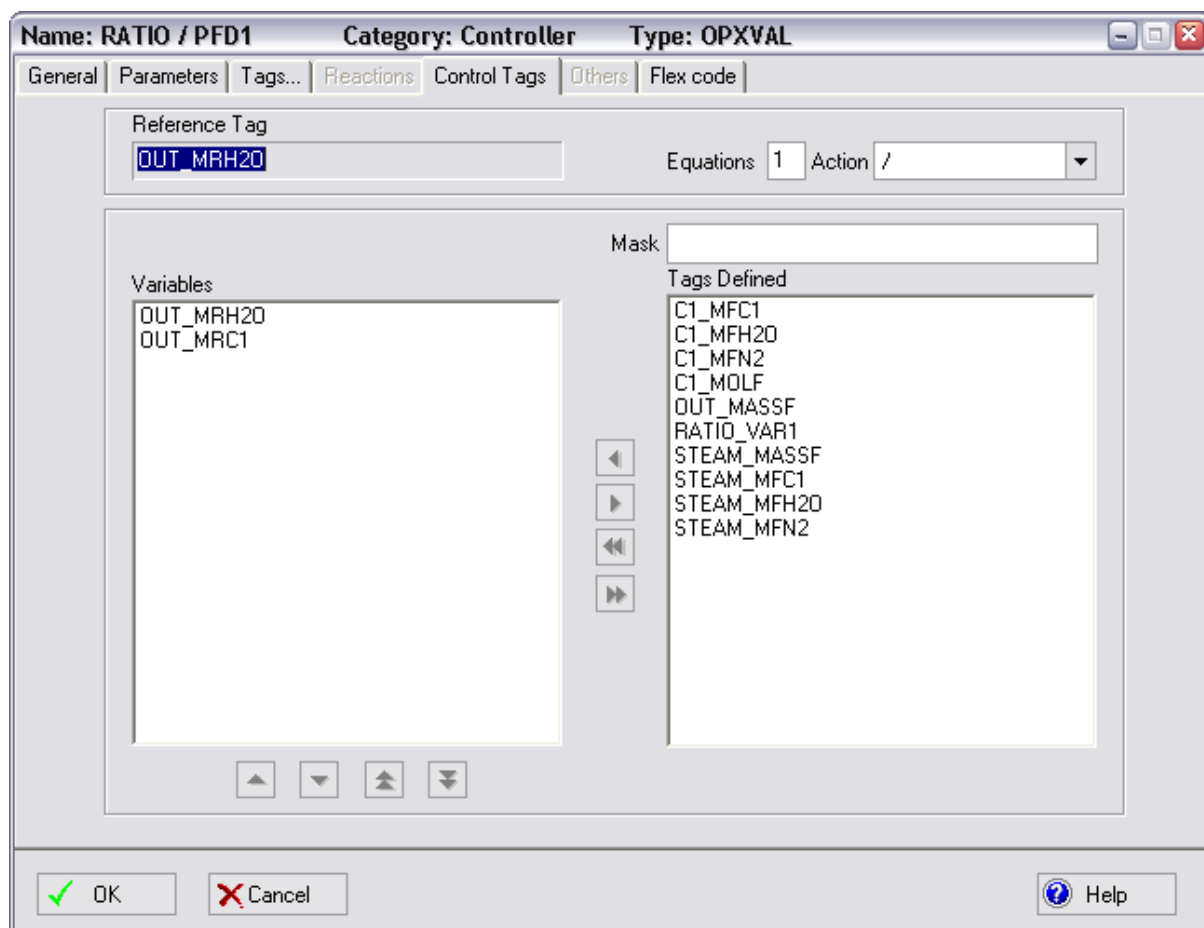


FIGURE 45 - THE CONTROL TAGS PAGE OF AN OPXVAL UNIT

VARi (Parameters)

Results of the equations ($i=1..9$) (type: Measurable).

EXAMPLE : SETTING STEAM TO METHANE MOLAR RATIO

Attention: see model opxval.bls in the directory "examples\opxval".

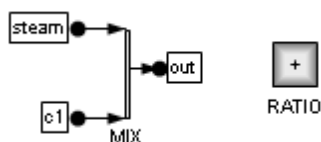


FIGURE 46 - AN EXAMPLE WITH OPXVAL TYPE UNIT

This example deals with a mixer of methane and steam for which it is requested that the molar ratio of steam to methane is equal to a specific value.

There are 2 UNITS:

- MIX:
 - type=BBXVAL
 - MODE='Mass Balance onl'y
 - MODEDP='None'
- RATIO:
 - type=OPXVAL
 - Equations: 1
 - Action: division (/)
 - Controlled variables: OUT_MRH2O, OUT_MRC1

3 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
STEAM	MIX	to MIX
C1	MIX	to MIX
OUT	MIX	from MIX

The THERMOD MIX can be derived from the THERMOD AIR available in Empty.bls and contains: C1, N2 and H2O.

The list of measurements and results are given in tables below.

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
C1_MFC1	STREAM	C1	MFC1	93.000	-1.000	%
C1_MFH2O	STREAM	C1	MFH2O	0.0000	CST	%
C1_MFN2	STREAM	C1	MFN2	7.0000	-0.1000	%
C1_MOLF	STREAM	C1	MOLF	50000.	-5.000	Nm3/h
OUT_MASSF	STREAM	OUT	MASSF	117.00	-5.000	t/h
OUT_MRC1	STREAM	OUT	C1		OFF	kmol/s
OUT_MRH2O	STREAM	OUT	H2O		OFF	kmol/s
RATIO_VAR1	UNIT	RATIO	VAR1	1.9000	-5.000	-
STEAM_MASSF	STREAM	STEAM	MASSF	80.000	-5.000	t/h
STEAM_MFC1	STREAM	STEAM	MFC1	0.0000	CST	%
STEAM_MFH2O	STREAM	STEAM	MFH2O		OFF	%
STEAM_MFN2	STREAM	STEAM	MFN2	0.0000	CST	%

TABLE 27 - MEASUREMENT LIST FOR OPXVAL EXAMPLE

Number of tags	12					
Number of equations	10					
Number of unmeasured variables	7					
Number of measured variables	6					
Number of variables considered constant	6					
Obvious number of redundancies	3					
Total number of redundancies	3					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	1.94365					
Chi-square test value	7.81000					
Sum of squares of residues	0.623330E-018					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
C1_MFC1	93.000	1.00 %	93.000	0.753E-02%	0.00	%
C1_MFH2O	0.0000	CST	0.0000			%
C1_MFN2	7.0000	0.100 %	7.0000	0.100 %	0.00	%
C1_MOLF	50000.	5.00 %	52099.	3.56 %	0.71	Nm3/h
OUT_MASSF	117.00	5.00 %	116.18	2.94 %	0.02	t/h
OUT_MRC1		OFF	0.60048	3.56 %		kmol/s
OUT_MRH2O		OFF	1.1863	3.44 %		kmol/s
RATIO_VAR1	1.9000	5.00 %	1.9756	3.94 %	0.63	-
STEAM_MASSF	80.000	5.00 %	76.940	3.44 %	0.59	t/h
STEAM_MFC1	0.0000	CST	0.0000			%
STEAM_MFH2O		OFF	100.00	0.00 %		%
STEAM_MFN2	0.0000	CST	0.0000			%

TABLE 28 - RESULTS OF OPXVAL EXAMPLE

SECTION 32: PIPEVAL : PIPELINES AND WELLS

PURPOSE

The PIPEVAL unit models:

- the pressure drop along lines and wells according to various equations (Blasious, Von Karmann, Colebrook). It also takes into account pressure changes according to the initial and final elevations of the pipe
- the heat transfer to an external media of given temperature profile

The trajectory of the pipe/well can be straight or not.

SYMBOLS

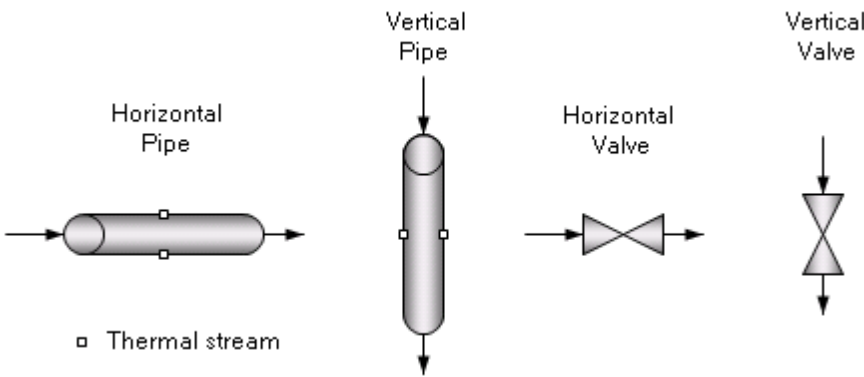


FIGURE 47 - SYMBOLS FOR UNIT PIPEVAL

CONNECTION RULES

INLET STREAMS

- 1 material STREAM

OUTLET STREAMS

- 1 material STREAM

Inlet and outlet STREAMS must use the same THERMOD.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account

NAME	TAB	TYPE	DESCRIPTION
MODEDP	General	Input	Pressure equation
MODEFRIC	General	Input	Friction Coefficient Equation
MODETOE Q	General	Input	Temperature equality
MODETRA N	General	Input	Heat Transfer Mode
PHASES	General	Input	Number of phases
TPROFILE	General	Input	Temperature Profile
TRAJECT	General	Input	Pipe Trajectory
MODETIME	General	Input	Residence Time Mode
MODERHO	General	Input	Average Rho Mode
MODECFD P	General	Input	Uncertainty on Pressure drop equation
DP	Parameters	Measurable	Pressure drop
DPX	Parameters	Input	Nominal pressure drop
MASSFx	Parameters	Input	Nominal mass flow rate
CFDP	Parameters	Measurable	DP correction factor
ZIN	Parameters	Input	Inlet Pipe Height
ZOUT	Parameters	Input	Outlet Pipe Height
DIAM	Parameters	Input	Internal Pipe Diameter
LENGTH	Parameters	Input	Pipe length
ROUGH	Parameters	Measurable	Pipe roughness
REYNOLDS	Parameters	Result	Reynolds
VISC	Parameters	Result	Viscosity

NAME	TAB	TYPE	DESCRIPTION
SPEED	Parameters	Measurable	Velocity
FRICTION	Parameters	Measurable	Friction coefficient
NSTEPS	Parameters	Input	Number of steps for stepwise integration
U	Parameters	Measurable	Overall Heat Transfer coefficient
EFFICNTU	Parameters	Measurable	NTU Efficiency
TEXTIN	Parameters	Measurable	Outside media Temperature at pipe inlet
TEXTOUT	Parameters	Measurable	Outside media Temperature at pipe outlet
DTEXT	Parameters	Measurable	Outside media Temperature gradient
RESTIME	Parameters	Measurable	Residence Time
RHO	Parameters	Result	Average density
DPGRAV	Parameters	Measurable	Gravity Pressure drop
DPFRIC	Parameters	Measurable	Friction Pressure drop
CFDGRAV	Parameters	Measurable	DP Gravity correction factor
CFDPFRIC	Parameters	Measurable	DP Friction correction factor
TFLOWi	Others	Intermediate	Temperature
PFLOWi	Others	Intermediate	Pressure

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Mass and Energy Balance	Mass and energy balances are taken into account
1	Mass Balance only	Only mass balances are taken into account

MODEDEP (General)

Pressure equation (type: Input)

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form: $DP = CFDP * DPX * (MASSF / MASSFX)^2$ where: ■ CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%) ■ DPX: the nominal pressure drop ■ MASSFX: the nominal flow rate ■ MASSF: the actual flow rate
3	DP=F(MASSF,friction)	a pressure drop equation will be generated and the pressure drop will be a function of the flow rate according to flow conditions (laminar, turbulent, etc.): $DP = CFDP * \left(\frac{Z_{out} - Z_{in}}{10^5} * \rho * g * CFDP_{grav} + \frac{f * \rho * v^2 * L}{10^5} * CFDP_{fric} \right)$ where: ■ Z_{in}: the elevation at pipe inlet ■ Z_{out}: the elevation at pipe outlet ■ ρ: the stream density [kg/m³] ■ g: the acceleration of gravity = 9.81 m/s² ■ f: the friction factor (see below) ■ v: the stream velocity [m/s] ■ L: the pipe length [m] ■ D: the pipe diameter [m]
4	DP=F(MASSF,Friction,...) Stepwise	a pressure drop equation will be generated and the pressure drop will be a function of the flow rate. The pressure drop will be calculated for each section of the pipe (NSTEPS). This mode is only valid if the unit is heat and mass balance (MODE = 0) and MODETRAN = 3

MODEFRIC (General)

Friction Coefficient Equation (type: Input).

0	Constant Friction Coefficient	Friction coefficient defined manually
1	Laminar Flow (Re < 2000)	$f = \frac{64}{Re}$
2	Turbulent (2000 < Re < 70000)	Flow $f = 0.3164 * Re^{(-0.25)}$ (Blasius)
3	Turbulent (70000 < Re < 1E27)	Flow $\frac{1}{\sqrt{f}} = 2 * \log_{10} \left(\frac{Re * \sqrt{f}}{2.51} \right)$ (Von Karman)
4	Turbulent Flow Roughness	with $\frac{1}{\sqrt{f}} = -2 * \log \left(\frac{2.51}{Re * \sqrt{f}} + \frac{\left(\frac{\varepsilon}{d} \right)}{3.72} \right)$ (Colebrook)

MODETOEQ (General)

Temperature equality (type: Input)

0	None	No additional equation will be generated
2	Inlet and outlet streams	The inlet and outlet streams will have the same temperature

MODETRAN (General)

Heat Transfer Mode (type: Input).

Heat transfer equations are only available when the unit is calculated mass and energy balance. Three different methods are available.

0	None	No additional equation will be generated
1	NTU method	Flat temperature of external media (TEXTIN). The heat load is calculated using the Number of Transfer Units method. The NTU method is based on the notion of efficiency, the efficiency being defined by the ratio between the actual heat load and the maximum heat load achievable assuming an infinite heat transfer surface area, that is when the stream reaches the temperature of the outside media.

$$\text{LOAD} = \text{EFFICNTU} \cdot \text{LOADMAX}$$

The efficiency is itself related to the heat transfer coefficient by:

$$\text{EFFICNTU} = 1 - e^{\left(\frac{-UA}{C}\right)}$$

where:

- U: the heat transfer coefficient [kW/m².K]
- A: the pipe inside area [m²]
- C: the total heat capacity of the stream (Massf * Cp) [kW/K]

- 2 Stepwise integration (constant Cp on each step) The pipe is divided into a number of steps of equal length (parameter NSTEPS). It is assumed that the pressure profile inside the pipe is linear versus pipe length and that there is no phase change in the pipe.

The heat transfer is integrated step by step from pipe inlet to pipe outlet according to the following formula:

$$H_{i+1} - H_i = C \cdot (T_{i+1} - T_i) + \frac{\partial H}{\partial P} DP = U DA (T_{ext} - 0.5(T_i + T_{i+1}))$$

which can be rewritten as:

$$T_{i+1} = \frac{CT_i + UDA(T_{ext} - 0.5T_i) - \frac{\partial H}{\partial P} DP}{C + 0.5 UDA}$$

where:

- H_i and H_{i+1} are the enthalpy flow at the inlet and outlet of step i [kW]
- T_i and T_{i+1} are the temperature at the inlet and outlet of step i
- $DP = (P_{out} - P_{in})/NSTEPS$ is the incremental pressure difference on a step

- $\frac{\partial H}{\partial P}$ reflects the impact of pressure on enthalpy
- C is the average heat capacity over the step (average of $Mass \cdot C_p$ between inlet and outlet conditions of the step) [kW/K]
- DA is the incremental heat exchange area per step [m²]:

$$DA = \pi \cdot DIAM \cdot LENGTH / NSTEPS$$

- T_{ext} is the average external media temperature in the step
- U is the heat transfer coefficient [kW/m².K]

3 Stepwise integration (rigorous) The pipe is divided into a number of steps of equal length (parameter NSTEPS).

The heat transfer is integrated step by step from pipe inlet to pipe outlet according to the following formula:

$$H_{i+1} - H_i = U \cdot DA \cdot (T_{ext} - 0.5(T_i + T_{i+1}))$$

where:

- H_i and H_{i+1} are the enthalpy flow at the inlet and outlet of step i [kW]
- T_i and T_{i+1} are the temperature at the inlet and outlet of step i
- C is the average heat capacity over the step (average of $Massf \cdot C_p$ between inlet and outlet conditions of the step) [kW/K]
- DA is the incremental heat exchange area per step [m²]:

$$DA = \pi \cdot DIAM \cdot LENGTH / NSTEPS$$

- T_{ext} is the average external media temperature in the step
- U is the heat transfer coefficient [kW/m².K]

The relation between temperature and enthalpy is rigorously calculated at each step during the integration.

PHASES (General)

Number of phases (type: Input).

This parameter is only used when MODETRAN=3. It indicates which phases have to be considered when calculating the enthalpy from temperature and pressure.

0	According to stream state	state inside the pipe is the same as stream state
1	V	stream is considered as vapour inside the pipe
2	L	stream is considered as liquid inside the pipe
4	LV (2 phases flow)	stream is considered as liquid/vapour inside the pipe
5	LL (2 phases flow)	stream is considered as liquid/liquid inside the pipe
6	LLV (3 phases flow)	stream is considered as liquid/liquid/vapour inside the pipe
10	WL (2 phases flow)	stream is considered as water/liquid inside the pipe
11	WLV (3 phases flow)	stream is considered as water/liquid/vapour inside the pipe

TPROFILE (General)

Temperature Profile (type: Input).

TPROFILE defines the shape of the temperature profile of the external media.

0	Constant Temperature	the external media is given by TEXTIN
1	Linear Profile versus Depth	the external media is interpolated between TEXTIN and TEXOUT
2	Profile given through a table	the temperature profile is given with the trajectory

TRAJECT (General)

Pipe Trajectory (type: Input).

0	Straight	the pipe is straight (but not necessarily horizontal or vertical)
1	Given through a table	the pipe coordinates are given through an external file

The table used to define pipe trajectory and/or temperature profile must be entered through a flat file which is referenced in a mea file through the syntax:

TABINPUT SOMEFILE.TXT

where somefile.txt is the name of the flat file.

Example of a trajectory file without temperature profile:

```
UNITNAME      L01-WELL-L      ! Name of the PIPEVAL unit
NUMTABLE      1              ! Table ID
MAXCOL 2      ! 2 columns (x,z) are given
PHYSUN LG m   LG m          ! Physical units for x and z
300.0 -2349.5 ! x, z coordinates
186.0 -2463.4
43.0  -2606.3
0.0   -2649.3
```

Example of a trajectory file with temperature profile:

```
UNITNAME      L01-WELL-L      ! Name of the PIPEVAL unit
NUMTABLE      1              ! Table ID
MAXCOL 3      ! 3 columns (x,z,T) are given
PHYSUN LG m   LG m   TE C    ! Physical units for x and z
0.0   0.0   24.4 ! x, z and T values
41.4  -39.9  20.4
122.4 -120.4 12.3
213.3 -210.8 4
306.0 -302.9 4
392.5 -388.8 4
```

MODETIME (General)

Residence Time Mode (type: Input).

0	Calculated after convergence residence time will be calculated after convergence is achieved
1	Calculated during convergence residence time is calculated at each iteration

MODERHO (General)

Average Rho Mode (type: Input).

0	Rho not variable of the problem Rho is not calculated at each iteration, value is calculated after convergence is achieved
1	Rho variable of the problem Rho is calculated at each iteration

MODECFDP (General)

Uncertainty on pressure drop equation (type: Input).

0	No uncertainty CFDP=1 constant
1	Global uncertainty (CFDP) CFDP variable
2	Uncertainty on Gravity and CFDPGRAV and CFDPFRIC variables Friction DP

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{in} - P_{out}$$

This equation is not generated if MODEDP = 0.

where:

- P_{in} : the pressure of the inlet stream
- P_{out} : the pressure of the outlet stream

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when MODEDP = 2.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

ZIN (Parameters)

Inlet Pipe Height (type: Input).

ZOUT (Parameters)

Outlet Pipe Height (type: Input).

DIAM (Parameters)

Internal Pipe Diameter (type: Input).

LENGTH (Parameters)

Pipe length (type: Input).

ROUGH (Parameters)

Pipe roughness (type: Measurable).

REYNOLDS (Parameters)

Reynolds (type: Result).

VISC (Parameters)

Viscosity (type: Result).

SPEED (Parameters)

Velocity (type: Measurable).

FRICTION (Parameters)

Friction coefficient (type: Measurable).

NSTEPS (Parameters)

Number of steps for stepwise integration (type: Input).

U (Parameters)

Overall Heat Transfer coefficient (type: Measurable).

EFFICNTU (Parameters)

NTU Efficiency (type: Measurable).

TEXTIN (Parameters)

Outside media Temperature at pipe inlet (type: Measurable).

TEXTOUT (Parameters)

Outside media Temperature at pipe outlet (type: Measurable).

DTEXT (Parameters)

Outside media Temperature gradient (type: Measurable).

$$DTEXT = (TEXTOUT - TEXTIN) / (ZOUT - ZIN)$$

RESTIME (Parameters)

Residence Time (type: Measurable).

RHO (Parameters)

Average density (type: Result).

DPGRAV (Parameters)

Gravity Pressure drop (type: Measurable).

DPFRIC (Parameters)

Friction Pressure drop (type: Measurable).

CFDPGRAV (Parameters)

DP Gravity correction factor (type: Measurable).

CFDPFRIC (Parameters)

DP Friction correction factor (type: Measurable).

TFLOWi (Others)

Temperature (type: Intermediate).

PFLOWi (Others)

Pressure (type: Intermediate).

EXAMPLE : PRESSURE DROP IN A LINE

Attention: see model pipeval.bls in the directory "examples\pipeval".

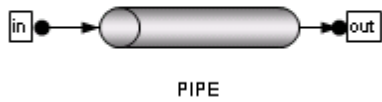


FIGURE 48 - PIPEVAL EXAMPLE

There is 1 UNIT:

- PIPE
 - type=PIPEVAL
 - MODE= 'Mass Balance only'
 - MODEDP= 'DP=F(MASSF,Friction,...) '
 - MODEFRIC=' Turbulent Flow with Roughness'
 - MODETOEQ= 'Temperature equality Inlet and Outlet streams '
 - DIAM= 0.1 m
 - LENGTH= 1000 m
 - ROUGH= 0.0001 m

2 STREAMS must be defined:

STREAMS THERMOD

IN WATER
OUT WATER

CONNECTIONS

to PIPE
from PIPE

The list of measurements is given in table below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
IN_COLDVOLF	STREAM	IN	COLDVOLF	1000.0	-3.000	m3/d
IN_P	STREAM	IN	P	100.00	-5.000	barg
IN_T	STREAM	IN	T	15.000	-5.000	C
OUT_COLDVOLF	STREAM	OUT	COLDVOLF		OFF	m3/d
OUT_P	STREAM	OUT	P		OFF	barg
PIPE_DP	UNIT	PIPE	DP		OFF	bar

TABLE 29 - MEASUREMENTS FOR PIPEVAL EXAMPLE

The results are (file r6v):

Number of tags	6					
Number of equations	7					
Number of unmeasured variables	7					
Number of measured variables	3					
Number of variables considered constant	0					
Obvious number of redundancies	0					
Total number of redundancies	0					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	0					
Chi-square test value	0					
Sum of squares of residues	0.459176E-012					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
IN_COLDVOLF	1000.0	3.00 %	1000.0	3.00 %	0.00	m3/d
IN_P	100.00	5.00 %	100.00	5.00 %	0.00	barg
IN_T	15.000	5.00 %	15.000	5.00 %	0.00	C
OUT_COLDVOLF		OFF	1000.0	3.00 %		m3/d
OUT_P		OFF	97.569	5.13 %		barg
PIPE_DP		OFF	2.4308	6.29 %		bar

TABLE 30 - RESULTS OF PIPEVAL EXAMPLE

SECTION 33: PRECIS: PRECISION OF CALCULATIONS

PURPOSE

The **PRECIS** unit allows to define precision on calculations.

SYMBOLS



FIGURE 49 - SYMBOLS FOR UNIT PRECIS

CONNECTION RULES

No connections are allowed on a **PRECIS**.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
KPRINT	(none)	Hidden	Printout level
EPSLV1	Parameters	Input	Tolerance for equations of state
EPSLV2	Parameters	Input	Tolerance for LVE at P,T (internal)
EPSLV3	Parameters	Input	Tolerance for LVE at P,T (external)
EPSLV4	Parameters	Input	Tolerance for LVE at P,VAPF or T,VAPF (internal)
EPSLV5	Parameters	Input	Tolerance for LVE at P,VAPF or T,VAPF (external)
EPSTHERM	Parameters	Input	Precision LVE at H or S (external)
EPSTRACE	Parameters	Input	Criteria of determination of trace
EPSLLV1	Parameters	Input	Precision LLVE criteria for K of 2 liquids
EPSLLV2	Parameters	Input	Precision LLVE convergence criteria
EPSLLV3	Parameters	Input	Precision LLVE stabilisation for composition

NAME	TAB	TYPE	DESCRIPTION
EPSLLV4	Parameters	Input	Precision LLVE at P,VAPF or T,VAPF (internal)
EPSLLV5	Parameters	Input	Precision LLVE at P,VAPF or T,VAPF (external)
EPSLLV6	Parameters	Input	LLVE initialization before resolution
EPSLLV7	Parameters	Input	Precision LLVE at T,P, calc. $V/L+V$ and $L1/L1+L2$
EPSLLV8	Parameters	Input	Precision LLVE at T,P, calc. $V/L+V$
EPSLLV9	Parameters	Input	Precision LLVE at T,P, calc. $L1/L1+L2$
EPCHM1	Parameters	Input	Precision molar flowrate in chemical equilibria
EPCHM2	Parameters	Input	Precision enthalpy in chemical equilibria
ITMLV1	Parameters	Input	Maximum number of iterations for thermodynamical calculation of volume
ITMLV2	Parameters	Input	Maximum number of iterations for alf computation between 0 and 1 (alf is the solution of $S(alf)=0$)
ITMLV3	Parameters	Input	Maximum number of iterations for isothermal flash calculation when there is more than 1 substance
ITMLV4	Parameters	Input	Maximum number of iterations for s(p/t) stabilization
ITMLV5	Parameters	Input	Maximum number of iterations for s(p/t)=1 resolution
ITMTHERM	Parameters	Input	Maximum number of iterations for h or s computation
ITMLL2	Parameters	Input	Maximum iterations
ITMLL4	Parameters	Input	Maximum iterations
ITMLL5	Parameters	Input	Maximum iterations
ITMLL6	Parameters	Input	Maximum iterations
ITMLL7	Parameters	Input	Maximum iterations

NAME	TAB	TYPE	DESCRIPTION
ITMLL8	Parameters	Input	Maximum iterations
ITMLL9	Parameters	Input	Maximum iterations
ITCHM1	Parameters	Input	Maximum iterations
ITCHM2	Parameters	Input	Maximum iterations
EPFABS1	Parameters	Input	Equilibria of plates (internal loop)
EPFABS2	Parameters	Input	Equilibria of plates (external loop)
XWAT0	Parameters	Input	Solubility of water in Oil
XWAT1	Parameters	Input	Solubility of water in Oil
WLVMODE	Parameters	Input	WLV mode

UNIT PARAMETERS

EPSLV1 (Parameters)

Tolerance for equations of state (type: Input).

EPSLV2 (Parameters)

Tolerance for LVE at P,T (internal) (type: Input).

EPSLV3 (Parameters)

Tolerance for LVE at P,T (external) (type: Input).

EPSLV4 (Parameters)

Tolerance for LVE at P,VAPF or T,VAPF (internal) (type: Input).

EPSLV5 (Parameters)

Tolerance for LVE at P,VAPF or T,VAPF (external) (type: Input).

EPSTHERM (Parameters)

Precision LVE at H or S (external)

EPSTRACE (Parameters)

Criteria of determination of trace

EPSLLV1 (Parameters)

Precision LLVE criteria for K of 2 liquids

EPSLLV2 (Parameters)

Precision LLVE convergence criteria (type:Input).

EPSLLV3 (Parameters)

Precision LLVE stabilisation for composition (type:Input).

EPSLLV4 (Parameters)

Precision LLVE at P,VAPF or T,VAPF (internal) (type:Input).

EPSLLV5 (Parameters)

Precision LLVE at P,VAPF or T,VAPF (external) (type:Input).

EPSLLV6 (Parameters)

LLVE initialization before resolution (type:Input).

EPSLLV7 (Parameters)

Precision LLVE at T,P, calc. $V/L+V$ and $L1/L1+L2$ (type:Input).

EPSLLV8 (Parameters)

Precision LLVE at T,P, calc. $V/L+V$ (type:Input).

EPSLLV9 (Parameters)

Precision LLVE at T,P, calc. $L1/L1+L2$ (type:Input).

EPCHM1 (Parameters)

Precision molar flowrate in chemical equilibria (type:Input).

EPCHM2 (Parameters)

Precision enthalpy in chemical equilibria (type:Input).

ITMLV1 (Parameters)

Maximum number of iterations for thermodynamical calculation of volume (type:Input).

ITMLV2 (Parameters)

Maximum number of iterations for alf computation between 0 and 1 (alf is the solution of $S(\text{alf})=0$) (type:Input).

ITMLV3 (Parameters)

Maximum number of iterations for isothermal flash calculation when there is more than 1 substance (type:Input).

ITMLV4 (Parameters)

Maximum number of iterations for s(p/t) stabilization (type:Input).

ITMLV5 (Parameters)

Maximum number of iterations for s(p/t)=1 resolution (type:Input).

ITMTHERM (Parameters)

Maximum number of iterations for h or s computation (type:Input).

ITMLL2 (Parameters)

Maximum iterations (type:Input).

ITMLL4 (Parameters)

Maximum iterations (type:Input).

ITMLL5 (Parameters)

Maximum iterations (type:Input).

ITMLL6 (Parameters)

Maximum iterations (type:Input).

ITMLL7 (Parameters)

Maximum iterations (type:Input).

ITMLL8 (Parameters)

Maximum iterations (type:Input).

ITMLL9 (Parameters)

Maximum iterations (type:Input).

ITCHM1 (Parameters)

Maximum iterations (type:Input).

ITCHM2 (Parameters)

Maximum iterations (type:Input).

EPFABS1 (Parameters)

Equilibria of plates (internal loop) (type:Input).

EPFABS2 (Parameters)

Equilibria of plates (external loop) (type:Input).

XWAT0 (Parameters)

Solubility of water in Oil (type:Input).

XWAT1 (Parameters)

Solubility of water in Oil (type:Input).

WLVMODE (Parameters)

WLV mode (type: Input).

1	1	Complete mode
2	2	Simplified mode (considers that water is not soluble in oil)

SECTION 34: PRHVAL : PREHEATER WITH CONDENSATION

PURPOSE

The **PRHVAL** unit allows you to model preheaters where a vapor stream (typically steam) is de-superheated, condensed and possibly sub-cooled to preheat a cold stream, without phase change. Three different zones are considered: de-superheating, condensing and sub-cooling, with different areas and heat transfer coefficients.

The sub-cooling zone can be merged with the condensing zone, in which case the possible sub-cooling is not evaluated. The outlet stream is assumed to be liquid, but the distance from the bubble point is not calculated.

SYMBOLS

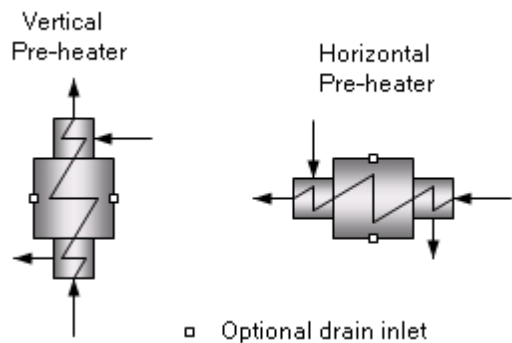


FIGURE 50 - SYMBOLS FOR UNIT PRHVAL

CONNECTION RULES

INLET STREAMS

- 2 or 3 material STREAMS

OUTLET STREAMS

- 2 material STREAMS

Corresponding inlet and outlet STREAMS must use the same THERMOD. The third inlet stream is optional and represents a drain inlet. The same THERMOD must be used for the first and the third inlet streams. The first inlet stream (going into the shell) is the hot vapor stream, while the cold stream goes through the tubes.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION

MODE	General	Input	Selection of equations to be taken into account
MODEDRDP	General	Input	Mode for drain pressure drop equation
MODESUB	General	Input	Mode for subcooling
TOTLOAD	Parameters	Measurable	Total heat load
TOTA	Parameters	Measurable	Total surface area
DRAINDP	Parameters	Measurable	Pressure drop at drain inlet
DRDPX	Parameters	Input	Nominal drain pressure drop
DRMASSFX	Parameters	Input	Nominal drain flow rate
DRCFDP	Parameters	Measurable	DP correction factor
DTSC	Parameters	Measurable	Level of sub-cooling
MODEDPi	Parameters	Input	Pressure equation
DPi	Parameters	Measurable	Pressure drop
DPXi	Parameters	Input	Nominal pressure drop
MASSFXi	Parameters	Input	Nominal flow rate
CFDPi	Parameters	Measurable	DP correction factor
LOAD1,2,3	Others	Measurable	Heat load of the 3 sections (de-superheating, condensing, sub-cooling)
A1,2,3	Others	Measurable	Surface area of the 3 sections
U1,2,3	Others	Measurable	Heat transfer coefficient in the 3 sections

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	Mass and Energy Balance	Mass and energy balances are taken into account
1	Mass Balance only	Only mass balances are taken into account
10	Performance equations	Mass, Energy and Performance equation (heat transfer coefficients)

The performance equation is for each of the three sections (1=desuperheating, 2=condensing, 3=subcooling) of the heat exchanger:

$$LOAD_i = A_i \cdot U_i \cdot LNTD_i$$

$$LNTD_i = \frac{DTSIDE1 - DTSIDE2}{\ln(DTSIDE1 / DTSIDE2)}$$

where:

- DTSIDE1 and DTSIDE2 are the temperature differences on each side of the section

MODEDRDP (General)

Mode for drain pressure drop equation (type: Input)

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DRDP = DRCFDP * DRDPX * (MASSF / DRMASSF) ^2$$

where:

- DRCFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%)
- DRDPX: the nominal pressure drop
- DRMASSF: the nominal flow rate
- MASSF: the actual flow rate of the stream at drain inlet

This parameter determines how to calculate the pressure drop between the drain inlet stream and the condensing zone pressure of the preheater.

MODESUB (General)

Mode for subcooling (type: Input).

0	With subcooling	Calculation of the subcooling zone
1	Without subcooling	No calculation of the subcooling

TOTLOAD (Parameters)

Total heat load (type: Measurable).

This is the total heat transferred in the three sections of the exchanger:

$$\text{TOTLOAD} = \text{LOAD1} + \text{LOAD2} + \text{LOAD3}$$

TOTA (Parameters)

Total surface area (type: Measurable).

This is the total surface area of the preheater:

$$\text{TOTA} = \text{A1} + \text{A2} + \text{A3}$$

DRAINDP (Parameters)

Pressure drop at drain inlet (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$\text{DRAINDP} = P_{\text{in}} - P_{\text{condensing zone}}$$

where:

- P_{in} : the pressure of the inlet stream
- $P_{\text{condensing zone}}$: the pressure of the outlet stream

This equation is not generated if $\text{MODEDP} = 0$.

DRDPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when $\text{MODEDP} = 2$.

DRMASSFX (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $\text{MODEDP} = 2$.

DRCFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $\text{MODEDP} = 2$.

DTSC (Parameters)

Level of sub-cooling (type: Measurable).

This parameter indicates the temperature difference between the outlet condensate and its saturation temperature.

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * (MASSF / MASSFX)^2$$

where:

- CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%)
- DPX: the nominal pressure drop
- MASSFX: the nominal flow rate
- MASSF: the actual flow rate

A value of MODEDP_i has to be given for each outlet streams.

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

LOAD1,2,3 (Others)

Heat load of the 3 sections (de-superheating, condensing, sub-cooling) (type: Measurable).

A1,2,3 (Others)

Surface area of the 3 sections

U1,2,3 (Others)

Heat transfer coefficient in the 3 sections (type: Measurable).

EXAMPLE : TWO CONDENSERS IN SERIES

Attention: see model `prhval.bls` in the directory "examples\prhval". Units of type PRHVAL are also used in the example `coal.bls` (coal fired power plant).

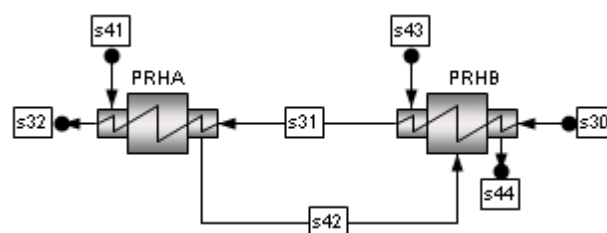


FIGURE 51 - PRHVAL EXAMPLE

There are 2 UNITS:

- PRHA,PRHB
 - type=PRHVAL
 - MODE='Performance equations'
 - MODESUB='With sub-cooling'
 - MODEDP1/2='DP variable'

7 STREAMS must be defined:

STREAMS	THERMOD	CONNECTIONS
S30	STEAM	to PRHB
S31	STEAM	from PRHB to PRHA
S32	STEAM	from PRHA
S41	STEAM	to PRHA
S42	STEAM	from PRHA to PRHB
S43	STEAM	to PRHB
S44	STEAM	from PRHB

The list of measurements is given in table below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
PRHA_DP1	UNIT	PRHA	DP1	0.50000E-01	-5.000	bar
PRHA_DP2	UNIT	PRHA	DP2	1.0000	-5.000	bar
PRHA_TOTA	UNIT	PRHA	TOTA	2828.4	CST	m2
PRHA_U1	UNIT	PRHA	U1	0.40000E-01	-10.00	kW/m2/K
PRHA_U2	UNIT	PRHA	U2	1.3000	-10.00	kW/m2/K
PRHA_U3	UNIT	PRHA	U3	0.15000	-10.00	kW/m2/K
PRHB_DP1	UNIT	PRHB	DP1	0.40000E-01	-5.000	bar
PRHB_DP2	UNIT	PRHB	DP2	1.0000	-5.000	bar
PRHB_TOTA	UNIT	PRHB	TOTA	3023.0	CST	m2
PRHB_U1	UNIT	PRHB	U1	0.40000E-01	-10.00	kW/m2/K
PRHB_U2	UNIT	PRHB	U2	1.3000	-10.00	kW/m2/K
PRHB_U3	UNIT	PRHB	U3	0.15000	-10.00	kW/m2/K
S30_MASSF	STREAM	S30	MASSF	183.09	-5.000	kg/s
S30_T	STREAM	S30	T	51.000	2.00	C
S31_MASSF	STREAM	S31	MASSF		OFF	kg/s
S31_P	STREAM	S31	P		OFF	bar
S31_T	STREAM	S31	T	84.000	2.00	C
S32_MASSF	STREAM	S32	MASSF		OFF	kg/s
S32_P	STREAM	S32	P	12.000	-5.000	bar
S32_T	STREAM	S32	T	114.00	3.00	C
S41_MASSF	STREAM	S41	MASSF	10.163	-5.000	kg/s
S41_P	STREAM	S41	P	1.9200	-5.000	bar
S41_T	STREAM	S41	T	195.00	2.00	C
S42_MASSF	STREAM	S42	MASSF		OFF	kg/s
S42_P	STREAM	S42	P	1.8700	-5.000	bar
S42_T	STREAM	S42	T	89.000	2.00	C
S43_MASSF	STREAM	S43	MASSF	9.6868	-5.000	kg/s
S43_P	STREAM	S43	P	0.63100	-5.000	bar
S43_T	STREAM	S43	T	99.000	2.00	C
S44_MASSF	STREAM	S44	MASSF		OFF	kg/s
S44_P	STREAM	S44	P	0.59099	-5.000	bar
S44_T	STREAM	S44	T	56.000	2.00	C

TABLE 31 - MEASUREMENTS FOR PRHV AL EXAMPLE

The results are (file r6v):

Number of tags	32
Number of equations	107
Number of unmeasured variables	101
Number of measured variables	25
Number of variables considered constant	11
Obvious number of redundancies	6
Total number of redundancies	6
Number of trivial redundancies	0

Validation and reconciliation results

Objective function	1.17499
Chi-square test value	12.6000
Sum of squares of residues	0.596303E-006
Number of active bounds	0
Number of variables close to bounds	0
Number of eliminated tags	0
Number of flagged tags	0
Number of filtered tags	0

TAGS IN THE MODEL

Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
PRHA_DP1	0.50000E-01	5.00 %	0.49993E-01	5.00 %	0.00	bar
PRHA_DP2	1.0000	5.00 %	1.0000	5.00 %	0.00	bar
PRHA_TOTA	2828.4	CST	2828.4			m2
PRHA_U1	0.40000E-01	10.0 %	0.40532E-01	9.43 %	0.02	kW/m2/K
PRHA_U2	1.3000	10.0 %	1.3133	9.64 %	0.01	kW/m2/K
PRHA_U3	0.15000	10.0 %	0.15072	9.89 %	0.00	kW/m2/K
PRHB_DP1	0.40000E-01	5.00 %	0.39981E-01	5.00 %	0.00	bar
PRHB_DP2	1.0000	5.00 %	1.0000	5.00 %	0.00	bar
PRHB_TOTA	3023.0	CST	3023.0			m2
PRHB_U1	0.40000E-01	10.0 %	0.40289E-01	9.84 %	0.01	kW/m2/K
PRHB_U2	1.3000	10.0 %	1.3166	9.61 %	0.02	kW/m2/K
PRHB_U3	0.15000	10.0 %	0.15152	9.73 %	0.01	kW/m2/K
S30_MASSF	183.09	5.00 %	185.61	3.70 %	0.08	kg/s
S30_T	51.000	2.00	50.443	1.44	0.08	C
S31_MASSF		OFF	185.61	6.86		kg/s
S31_P		OFF	13.000	0.602		bar
S31_T	84.000	2.00	82.425	0.866	0.62	C
S32_MASSF		OFF	185.61	3.70 %		kg/s
S32_P	12.000	5.00 %	12.000	5.00 %	0.00	bar
S32_T	114.00	3.00	114.11	1.19	0.00	C
S41_MASSF	10.163	5.00 %	9.9539	3.91 %	0.17	kg/s
S41_P	1.9200	5.00 %	1.9312	3.22 %	0.01	bar
S41_T	195.00	2.00	194.97	2.00	0.00	C
S42_MASSF		OFF	9.9539	3.91 %		kg/s
S42_P	1.8700	5.00 %	1.8812	3.30 %	0.01	bar
S42_T	89.000	2.00	89.194	1.95	0.01	C
S43_MASSF	9.6868	5.00 %	9.6033	4.39 %	0.03	kg/s
S43_P	0.63100	5.00 %	0.63602	3.08 %	0.03	bar
S43_T	99.000	2.00	98.901	1.99	0.00	C
S44_MASSF		OFF	19.557	2.93 %		kg/s
S44_P	0.59099	5.00 %	0.59604	3.28 %	0.03	bar
S44_T	56.000	2.00	56.423	1.85	0.04	C

TABLE 32 - RESULTS OF PRHVAL EXAMPLE

SECTION 35: PUMVAL : LIQUID PUMP

PURPOSE

The **PUMVAL** unit allows you to model pumps.

Measurements can be defined on the efficiency and the mechanical power as well as the distance from performance curves.

Note: When the Mechanical Stream is OFF, VALI will consider that the Material Stream go through the pump with a pressure drop equal to 0.

SYMBOLS

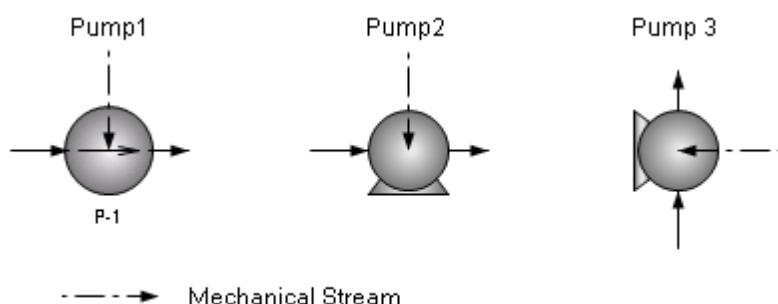


FIGURE 52 - SYMBOLS FOR UNIT PUMVAL

CONNECTION RULES

INLET STREAMS

- 1 material
- 1 mechanical

OUTLET STREAMS

- 1 material

Both material streams must use the same THERMOD.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account

NAME	TAB	TYPE	DESCRIPTION
MODEDP	General	Input	Pressure equation
MODEPERF	General	Input	Performance mode
EFFIC	Parameters	Measurable	Volumetric efficiency
DP	Parameters	Measurable	Pressure increase
PRATIO	Parameters	Measurable	Pressure ratio
RPMR	Parameters	Input	Reference pump speed
RPMMIN	Parameters	Input	Minimum performance speed
VOLFMIN	Parameters	Input	Minimum volumetric flow rate
NSTAGES	Parameters	Input	Number of stages
EFFi	Parameters	Input	Polynomial coefficients for efficiency curve
HEADi	Parameters	Input	Polynomial coefficients for head curve
DEFF	Parameters	Measurable	Distance from efficiency curve
CFEFF	Parameters	Measurable	Relative efficiency factor
EFFC	Parameters	Measurable	Efficiency from curve
DHEAD	Parameters	Measurable	Distance from head curve
DHEADR	Parameters	Measurable	Distance from head at nominal speed
HEAD	Parameters	Measurable	Actual head
HEADR	Parameters	Measurable	Reduced head (at nominal speed)
HEADC	Parameters	Result	Head from curve
HEADCR	Parameters	Result	Reduced head from curve (at nominal speed)

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	All equations	Mass and energy balances are taken into account. The volumetric efficiency equation is generated as well
1	Mass Balance only	Only mass balances are taken into account
2	Mass and Energy Balance	Mass and energy balances are taken into account

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

MODEPERF (General)

Performance mode (type: Input).

0	None	No performance equation will be generated
1	Efficiency	Efficiency curve will be generated
2	Head	Head curve will be generated
3	Efficiency & Head	Efficiency & head curves will be generated

EFFIC (Parameters)

Volumetric efficiency (type: Measurable).

This parameter is only relevant in MODE 0 and is defined by:

$$\text{POWER} = \frac{\text{VOLF} * (P_{\text{out}} - P_{\text{in}})}{\text{EFFIC}}$$

where:

- POWER: the pump power
- VOLF: the volumetric flow rate in the inlet conditions
- Pout and Pin: the inlet and outlet pressures

DP (Parameters)

Pressure increase (type: Measureable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{out} - P_{in}$$

This equation is not generated if MODEDP = 0.

PRATIO (Parameters)

Pressure ratio (type: Measurable)

$$PRATIO = P_{out}/P_{in}$$

This equation is not generated if MODEDP = 0.

RPMR (Parameters)

Reference pump speed (type: Input).

The reference pump speed corresponds to the pump speed for which the performance curves have been identified.

RPMMIN (Parameters)

Minimum performance speed (type: Input).

When the measured pump speed is below RPMMIN, the head curve equation is deactivated and the efficiency is calculated for the minimum rotation speed.

VOLFMIN (Parameters)

Minimum volumetric flow rate (type: Input).

VOLFMIN defines the lower bound of the validity range of the performance curves at the reference rotation speed. When the flow rate is below VOLFMIN, the efficiency and head are calculated at VOLFMIN.

NSTAGES (Parameters)

Number of stages (type: Input).

NSTAGES is used to calculate the overall head curve of a pump made of several stages.

EFFi (Parameters)

Polynomial coefficients for efficiency curve (type: Input).

The theoretical efficiency of the pump is defined by the following curve:

$$EFFC = EFF0 + \sum_{i=1}^5 EFFi * VOLFR^i$$

where:

- VolFR is the reduced volumetric flow rate = VolF RPMR / RPM
- EFFC is the calculated efficiency from curve

HEADi (Parameters)

Polynomial coefficients for head curve (type: Input).

The theoretical head of the pump is defined by the following curve:

$$\text{HEADCR} = \text{NSTAGES} * \left(\text{HEAD0} + \sum_{i=1}^5 \text{HEAD}_i \text{VOLFR}^i \right)$$

where:

- VolFR is the reduced volumetric flow rate = VolF RPMR / RPM

DEFF (Parameters)

Distance from efficiency curve (type: Measurable).

This parameter represents the distance between the actual pump efficiency and the theoretical efficiency defined by the pump theoretical curve.

$$\text{DEFF} = \text{EFFIC} - \text{EFFC}$$

Typically, a tag will be defined on DEFF with an estimated value of 0 and an uncertainty corresponding to the allowed distance between the actual and theoretical efficiency.

CFEFF (Parameters)

Relative efficiency factor (type: Measurable).

This parameter represents the actual efficiency relative to curve efficiency, as the ratio of actual efficiency on curve efficiency:

$$\text{CFEFF} = \frac{\text{EFF}}{\text{EFFC}}$$

EFFC (Parameters)

Efficiency from curve (type: Measurable).

This parameter represents the efficiency calculated from theoretical curve. See EFFi above for more information.

DHEAD (Parameters)

Distance from head curve (type: Measurable).

This parameter represents the distance between the actual pump head and theoretical head defined by the pump theoretical curve.

$$DHEAD = HEAD - HEADC$$

where:

- g is the gravity acceleration (9.81 m/s^2)
- ρ is the outlet stream density (kg/m^3)

Typically, a tag will be defined on DHEAD with an estimated value of 0 and an uncertainty corresponding to the allowed distance between the actual and theoretical efficiency.

DHEADR (Parameters)

Distance from head at nominal speed (type: Measurable).

This parameter represents the difference between head at nominal speed and head from curve at nominal speed:

$$DHEADR = HEADR - HEADCR$$

HEAD (Parameters)

Actual head (type: Measurable).

This parameter represents the actual head, calculated as:

$$HEAD = \frac{10^5 DP}{\rho g}$$

HEADR (Parameters)

Reduced head (at nominal speed) (type: Measurable).

This parameter represents the reduced head, i.e. the actual head at nominal speed.

$$HEADR = HEAD * \left(\frac{RPMR}{RPM} \right)^2$$

HEADC (Parameters)

Head from curve (type: Result).

This parameter represents the head from theoretical curve, at actual speed.

$$HEADC = HEADCR * \left(\frac{RPM}{RPMR} \right)^2$$

HEADCR (Parameters)

Reduced head from curve (at nominal speed) (type: Result).

This parameter represents the head calculated from the theoretical curve, at nominal speed. See

HEADi above for more information.

EXAMPLE : BOILER FEED WATER PUMP

Attention: see model `pumval.bls` in the directory "examples\pumval". Units of type PUMVAL are also used in the example `coal.bls` (coal fired power plant).

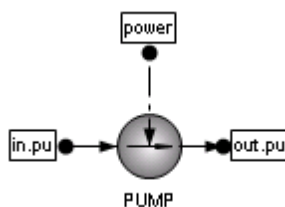


FIGURE 53 - PUMVAL EXAMPLE

There is 1 UNIT:

- PUMP
 - type=PUMVAL
 - MODE='All equations'
 - MODEDP='None'

3 STREAMS must be defined:

STREAMS	THERMOD	CONNECTIONS
IN.PU	STEAM	to PUMP
POWER	MECHANICAL STREAM	to PUMP
OUT.PU	STEAM	from PUMP

The THERMOD STEAM is defined in the file `Empty.bls`.

The list of measurements and results are given in tables below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
IN.PU_MASSF	STREAM	IN.PU	MASSF	40.000	-5.000	t/h
IN.PU_P	STREAM	IN.PU	P	10.000	0.100	bar
IN.PU_T	STREAM	IN.PU	T	50.000	1.00	C
OUT.PU_P	STREAM	OUT.PU	P	100.00	0.500	bar
OUT.PU_T	STREAM	OUT.PU	T	50.500	1.00	C
POWER_POWER	STREAM	POWER	POWER	100.00	-0.5000	kW
PUMP_EFFIC	UNIT	PUMP	EFFIC	90.000	5.00	%

TABLE 33 - MEASUREMENTS FOR PUMVAL EXAMPLE

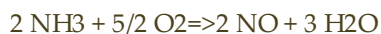
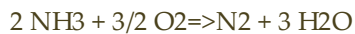
Number of tags	7					
Number of equations	5					
Number of unmeasured variables	3					
Number of measured variables	7					
Number of variables considered constant	0					
Obvious number of redundancies	2					
Total number of redundancies	2					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	2.44280					
Chi-square test value	5.99000					
Sum of squares of residues	0.346803E-012					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
IN.PU_MASSF	40.000	5.00 %	37.786	3.73 %	1.23	t/h
IN.PU_P	10.000	0.100	10.002	0.100	0.00	bar
IN.PU_T	50.000	1.00	50.028	0.706	0.00	C
OUT.PU_P	100.00	0.500	99.942	0.499	0.01	bar
OUT.PU_T	50.500	1.00	50.462	0.710	0.00	C
POWER_POWER	100.00	0.500 %	100.05	0.499 %	0.01	kW
PUMP_EFFIC	90.000	5.00	95.455	3.57	1.19	%

TABLE 34 - RESULTS OF PUMVAL EXAMPLE

SECTION 36: RCT-HNO3 : NH3 OXIDATION

PURPOSE

The **RCT-HNO3** unit models the conversion of NH3 according to the following reactions:



The conversion of the NH3 is complete. The percentage of NH3 which is converted to NO (according to the second reaction) is defined by the **SELECT** parameter (selectivity).

SYMBOLS

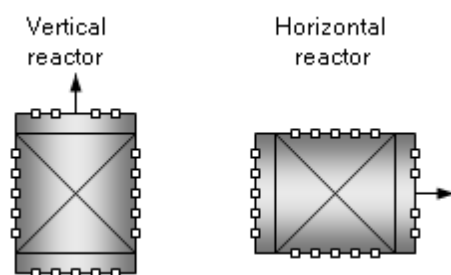


FIGURE 54 - SYMBOLS FOR UNIT RCT-HNO3

CONNECTION RULES

INLET STREAMS

- 1 material (any state)

OUTLET STREAMS

- 1 material (state LV)

THERMOD

The Thermod of the inlet stream must contain, in any order, at least the following compounds: NH3, O2, N2, H2O.

The Thermod of the outlet stream must contain, in any order, at least the following compounds: NO, NO2, N2O4, O2, N2, H2O, HNO3.

REACTION

The reactions are part of the model and do not need to be defined in the Reactions page.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure drop balance
SELEC	Parameters	Measurable	Selectivity
DP	Parameters	Measurable	Pressure drop
DPX	Parameters	Input	Nominal pressure drop
MASSEFX	Parameters	Input	Nominal flow rate
CFDP	Parameters	Measurable	DP correction factor
REFDP	Parameters	Input	Reference stream for pressure drop
EXTENTi	Parameters	Measurable	Extents of the reactions

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	Atomic and Energy Balance	Atomic and energy balances are taken into account
1	Atomic Balance only	Only atomic balances are taken into account

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form: $DP = CFDP * DPX * (MASSF / MASSFX)^2$ where: ■ CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%) ■ DPX: the nominal pressure drop ■ MASSFX: the nominal flow rate ■ MASSF: the actual flow rate

SELEC (Parameters)**Selectivity (type: Measurable).**

The selectivity is defined as the percentage of NH₃ which is converted to NO.

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when MODEDP = 2.

MASSFX (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

EXTENT1,2 (Parameters)

Extents of the reactions (type: Measurable).

The Extents are defined (on molar basis) as follows:

Extent1: amount of N_2 produced

Extent2: two times the amount of NO produced

EXAMPLE : NITRIC ACID PROCESS

Attention: see model `nitric.bls` in the directory "`\examples\nitric`."

SECTION 37: RCTVAL : REACTOR WITH DEFINED REACTIONS

PURPOSE

The **RCTVAL** unit allows you to model reactors and takes the atomic balances, energy balance and pressure drop into account.

Contrary to the **BATVAL** unit, the chemical reactions must be defined. All reactions are considered reversible.

SYMBOLS

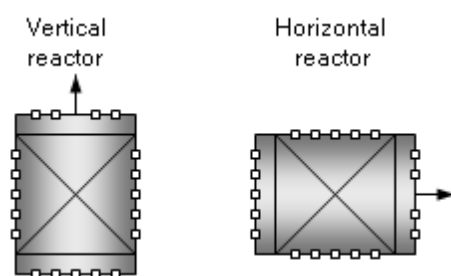


FIGURE 55 - SYMBOLS FOR UNIT RCTVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 10
- Type: material (at least one), thermal, mechanical

OUTLET STREAMS

- Maximum number: 10
- Type: one material as the first outlet (as shown on the symbols), thermal, mechanical

THERMOD

All chemical compounds involved in the chemical reactions must be present in the outlet THERMOD.

REACTIONS

- Maximum number: 30

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure drop balance
DP	Parameters	Measurable	Pressure drop
DPX	Parameters	Input	Nominal pressure drop
MASSFx	Parameters	Input	Nominal flow rate
CFDP	Parameters	Measurable	DP correction factor
REFDP	Parameters	Input	Reference stream for pressure drop
EXTENTi	Reactions	Measurable	Extents of the reactions

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	Atomic and Energy Balance	Atomic and energy balances are taken into account
1	Atomic Balance only	Only atomic balances are taken into account

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form: $DP = CFDP * DPX * (MASSF / MASSFX)^2$ where: ■ CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%) ■ DPX: the nominal pressure drop ■ MASSFX: the nominal flow rate ■ MASSF: the actual flow rate

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

EXTENTi (Reactions)

Extents of the reactions (type: Measurable).

The EXTENT of a reaction is the amount of reference product (with stoichiometric coefficient equal to 1) produced in the associated reaction.

For a particular component, the amount produced in a given reaction is the result of the multiplication of the EXTENT of the reaction by the stoichiometric coefficient of the component (thus value is negative for reagents).

EXAMPLE : NH₃ SYNTHESIS REACTOR

Attention: see model `rctval.bls` in the directory "examples\rctval".

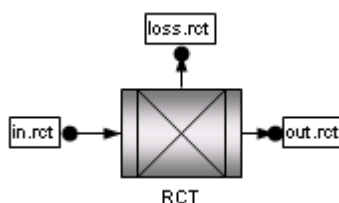


FIGURE 56 - AN EXAMPLE OF RCTVAL TYPE UNIT

This example deals with a chemical reactor for the synthesis of NH₃.

There is 1 UNIT:

- SYNT
 - type=RCTVAL
 - MODE='All Equations'
 - MODEDP='DP variable'

The reaction SYNT must be defined as: $3 \text{ H}_2 + 2 \text{ N}_2 \rightleftharpoons 2 \text{ NH}_3$

3 STREAMS must be defined:

STREAMS THERMOD

CONNECTIONS

IN.RCT RCMOD

to SYNT

OUT.RCT RCMOD

from SYNT

LOSS.RCT THERMAL STREAM

from SYNT

The THERMOD RCMOD can be derived from the THERMOS AIR available in Empty.bls and contains: H2, N2, NH3 and Ar.

The list of measurements is:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
IN.RCT_MFAR	STREAM	IN.RCT	MFAR	5.7000	-5.000	%
IN.RCT_MFH2	STREAM	IN.RCT	MFH2	70.000	-5.000	%
IN.RCT_MFN2	STREAM	IN.RCT	MFN2	21.000	-5.000	%
IN.RCT_MFNH3	STREAM	IN.RCT	MFNH3	3.3000	-5.000	%
IN.RCT_MOLF	STREAM	IN.RCT	MOLF	0.78000E+06	-5.000	Nm3/h
IN.RCT_P	STREAM	IN.RCT	P	200.00	-2.000	bar
IN.RCT_T	STREAM	IN.RCT	T	282.00	1.00	C
OUT.RCT_MFAR	STREAM	OUT.RCT	MFAR	6.0000	-5.000	%
OUT.RCT_MFH2	STREAM	OUT.RCT	MFH2	61.000	-5.000	%
OUT.RCT_MFN2	STREAM	OUT.RCT	MFN2	17.000	-5.000	%
OUT.RCT_MFNH3	STREAM	OUT.RCT	MFNH3	16.000	-5.000	%
OUT.RCT_P	STREAM	OUT.RCT	P	198.00	-2.000	bar
OUT.RCT_T	STREAM	OUT.RCT	T	470.00	2.00	C
RCT_DP	UNIT	RCT	DP	1.5000	-20.00	bar

TABLE 35 - MEASUREMENT LIST FOR RCTVAL EXAMPLE

And the results are (r6v):

Number of tags	14					
Number of equations	17					
Number of unmeasured variables	12					
Number of measured variables	14					
Number of variables considered constant	0					
Obvious number of redundancies	5					
Total number of redundancies	5					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	0.95623					
Chi-square test value	11.0800					
Sum of squares of residues	0.154187E-014					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
IN.RCT_MFAR	5.7000	5.00 %	5.5114	3.55 %	0.44	%
IN.RCT_MFH2	70.000	5.00 %	70.449	0.902 %	0.02	%
IN.RCT_MFN2	21.000	5.00 %	20.738	2.89 %	0.06	%
IN.RCT_MFNH3	3.3000	5.00 %	3.3023	4.99 %	0.00	%
IN.RCT_MOLF	0.78000E+06	5.00 %	0.78000E+06	5.00 %	0.00	Nm3/h
IN.RCT_P	200.00	2.00 %	199.75	1.41 %	0.00	bar
IN.RCT_T	282.00	1.00	282.00	1.00	0.00	C
OUT.RCT_MFAR	6.0000	5.00 %	6.1863	3.53 %	0.39	%
OUT.RCT_MFH2	61.000	5.00 %	60.707	1.59 %	0.01	%
OUT.RCT_MFN2	17.000	5.00 %	17.154	3.92 %	0.03	%
OUT.RCT_MFNH3	16.000	5.00 %	15.952	4.80 %	0.00	%
OUT.RCT_P	198.00	2.00 %	198.25	1.42 %	0.00	bar
OUT.RCT_T	470.00	2.00	470.00	2.00	0.00	C
RCT_DP	1.5000	20.0 %	1.5014	20.0 %	0.00	bar

TABLE 36 - RESULTS OF RCTVAL EXAMPLE

SECTION 38: REPVAL : REPORT SETTINGS

PURPOSE

The **REPVAL** unit allows controlling the amount and the type of information printed out in the main report file (r4v).

SYMBOLS



FIGURE 57 - SYMBOL FOR UNIT REPVAL

CONNECTION RULES

No connections are allowed on a **REPVAL**.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
SEQREP	General	Input	Report on Sequence
GLOBBAL	General	Input	Global balance
UNIPAR	General	Input	Unit parameters
UNISTR	General	Input	Unit inlets/outlets
STRREP	General	Input	Streams
TAGREP	General	Input	Tags
REACREP	General	Input	Reactions
THERREP	General	Input	Thermods
COMPREP	General	Input	Compounds
VARREP	General	Input	Variables/Constants
BNDREP	General	Input	Variable Bounds

NAME	TAB	TYPE	DESCRIPTION
COBAREP	General	Input	Compounds Balance
INDREP	General	Input	Indexes at the bottom
IMAGE	General	Input	Images
ORDVARE P	General	Input	Order of unmeasured variables
ZONEBAL	General	Input	Balances by zone
ORDTAREP	General	Input	Order of Tags

UNIT PARAMETERS

SEQREP (General)

Report on Sequence (type: Input).

0	0: (none)	No sequence report
1	1: Yes	The sequence report is the list of units that are in the current model (only with text reports or with the -HTO runtime option):

MAIN	(	LOFUNIT	)	
OPTVAL	(	OPTVAL	)	
GT	(	LOFUNIT	)	
	COMP	(	COMVAL	)
	SUMMER	(	EATVAL	)
	TUNE	(	TUNVAL	)
	POWER	(	BENVAL	)
REPVAL	(	REPVAL	)	

GLOBBAL (General)

Global balance (type: Input).

0 0: (none)

No global balance report

1 1: Yes

The global balance report summarizes the mass balance of the complete flow-sheet, presenting all the material inlets and outlets of the model. Tanks inventory changes are included:

GLOBAL BALANCE			
Streams	Units	Molar Flowrate (kmol/s)	Mass Flowrate (kg/s)
inlet streams:			
1	E-2	10.7118	308.452
2	E-4	1.91708	23.0364
CD.3	P-1	12.2471	220.636
CW.0	P-3	440.080	7928.21
SUM OF ALL INLETS:		464.956	8480.34
OUTLET STREAMS:			
CD.2	COND	12.2471	220.636
CW.2	COND	440.080	7928.21
FUMES.8	FUMES	10.9921	331.489
SUM OF ALL OUTLETS:		463.319	8480.34
BALANCE		1.63678	0.559751E-005

2 2: Balances by PFD

In addition to the global balance report, a similar report is issued for each PFD of the model

UNIPAR (General)

Unit parameters (type: Input).

0 0: (none)

No report on unit parameters

1 1: Yes

The Unit parameters report includes all the parameters of the units:

R-1 (unit type: BATVAL)			
UNIT PARAMETERS			
Equations	Atomic and Energy Balance		
Pressure equation	DP variable		
Reference stream for dp AIR.2	0		
Pressure drop	0.999969E-001 bar		R-1_DP

UNISTR (General)

Unit inlets/outlets (type: Input).

- 00: (none)No report on unit inlet/outlet streams
- 11: Streams Mass BalanceIncludes the measured and validated mass flow rates of the streams connected to the unit:

Stream	Unit	Tag	Mea. Val.	Rec. Val.	Unit
INLET STREAMS					
AIR.2	C-1		[66.54]	66.541	kg/s
NG	R-1	NG_MOLF	1.1930	1.2000	kg/s
OUTLET STREAMS					
FUME.1	T-1		[67.74]	67.741	kg/s
UNIT BALANCE			-707338E-002	0.570731E-006	kg/s

- 22: Streams Thermal BalanceReportIncludes the temperature, pressure, vapor fraction, mass flow rate and enthalpy flow of each connected stream:

Stream	Unit	Sta	T C	P bar	Vapf %	Massf kg/s	H kW
INLET STREAMS							
AIR.2	C-1	LV	499.93 Q	20.016 Q	100.000	66.541 Q	33435.
NG	R-1	LV	15.00 Q	20.016 Q	100.000	1.2000 Q	-26.363
OUTLET STREAMS							
FUME.1	T-1	LV	1202.53 Q	19.916	100.000	67.741 Q	93412.
HFORM CORRECTION							60002.9
BALANCE						0.570731E-006	-226569E-004

- 33: Streams in Molar flow ratesIncludes (in vertical table) the parameters and composition (molar flow rates) of each connected stream:

		***** * Inlet 1 * *****		***** * Inlet 2 * *****		***** * Outlet 1 * *****	
Stream		AIR.2		GN		FUME.1	
Type/State		MAT/LV		MAT/LV		MAT/LV	
From		COMP				BURNER	
To		BURNER		BURNER		TURB	
Thermod		AIR		GAS		FUMES	
T	C	499.856 ▶		15.0020 ▶		1198.36 ▶	
P	bar	20.0650		20.0000		19.9650	
VAPF	%	100.000		100.000		100.000	
MIXF	%	100.000		100.000		100.000	
H	kJ/kmol	14494.0		-352.389		39069.7	
S	kJ/kmol/C	8.30863		-26.0002		33.2809	
A	kJ/kmol	12016.8		7399.56		29147.0	
V	m3/kmol	3.20309		1.19789		6.12769	
MOLW	kg/kmol	28.8503		16.0430		28.4506	
HW	kJ/kg	502.386		-21.9653		1373.25	
MOLF	kmol/s	2.35611		0.759138E-01 ▶		2.43202	
MASSEF	kg/s	67.9744		1.21788		69.1923	
O2	kmol/s	0.494782		...		0.342955	
N2	kmol/s	1.86132		0.00000 ▶		1.86132	
H2	kmol/s	...		0.00000		...	
H2S	kmol/s	...		0.00000		...	
CO2	kmol/s	...		0.00000		0.759138E-01	
C1	kmol/s	...		0.759138E-01 ▶		...	
C2	kmol/s	...		0.00000 ▶		...	
C3	kmol/s	...		0.00000		...	
I-C4	kmol/s	...		0.00000 ▶		...	
N-C4	kmol/s	...		0.00000		...	
N-C5	kmol/s	...		0.00000		...	
H2O	kmol/s	...		...		0.151828	
SO2	kmol/s	...		...		0.00000	

- 44: Streams in Mass flow ratesIncludes (in vertical table) the parameters and composition (mass flow rates) of each connected stream.
- 55: Streams in Molar fractionsIncludes (in vertical table) the parameters and composition (molar fractions) of each connected stream (only with text reports or with the –HTO runtime option)
- 66: Streams in Mass fractionsIncludes (in vertical table) the parameters and composition (mass fractions) of each connected stream (only with text reports or with the –HTO runtime option)
- 77: Streams Thermal Balance(HFLOW)Similar as 1 but the enthalpy flow is based on HFLOW and not on the enthalpy flow by reference to the ideal gas at 25°C:

Stream	Unit	Sta	T C	P bar	Vapf %	Massf kg/s	HFLOW kW
INLET STREAMS							
AIR.2	C-1	LV	499.93 Q	20.016 Q	100.000	66.541 Q	33435.
NG	R-1	LV	15.00 Q	20.016 Q	100.000	1.2000 Q	-26.363
OUTLET STREAMS							
FUME.1	T-1	LV	1202.53 Q	19.916	100.000	67.741 Q	99991.
HFORM CORRECTION							60002.9
BALANCE						0.570731E-006	-6579.70

The information contained in this document is provided to Customers under a special license agreement. 601

document is not to be reproduced. It is provided to Customers under a special license agreement. 601

88: VDI-2048Includes the measured and validated mass flow rates and energy flows of the streams connected to the unit:

STRREP (General)

Streams (type: Input).

- 00: (none)
- 11: Full Report

No report on streams

For each stream, a complete report is printed with the compositions expressed in partial rates and fraction, on both molar and weight basis:

Stream: AIR.1

State: LV

Origine: C-1

Destination: AIR.1

Rates: AIR

Thermod: AIR

Name	Value	Unit	Tags
T	15.0492	C	AIR.1_T
P	1.01167	bar	AIR.1_P
VAPF	100	%	
MIXF	100	%	
H	-288.301	kJ/kmol	
S	3.30272	kJ/kmol/C	
A	-1273.01	kJ/kmol	
V	23.6854	m³/kmol	
MOLW	28.8503	kg/kmol	
HW	-9.99298	kJ/kg	
MOLF	0.186103E+006	Nm³/h	AIR.1_MOLF
MASSF	66.5413	kg/s	

Name	Molar flowrate	Molar fraction	Mass flowrate	Mass fraction	Tags
O2	39081.5	21.0000	15.4986	23.2918	AIR.1_MFO2
N2	0.147021E+006	79	51.0426	76.7082	
TOTAL FLOWRATE:	0.186103E+006	Nm³/h	66.5413	kg/s	AIR.1_MOLF

- 22: VDI-2048

For each stream, a complete report is printed with the measured and validated value and accuracy of all related variables. Whenever possible, the total mass flow rate and the enthalpy flow are estimated on basis of available measurements:

AIR.1	AIR	LV	From:	To: C-1	
	Mea.Val	95%Conf	Rec.Val	95%Conf	Unit
AIR.1_T	15.0000	1.00000	15.0492	0.994015	C
AIR.1_P	1.01300	0.202600E-01	1.01167	0.199102E-01	bar
VAPF			100.000	0.00000	%
H			-288.301	28.7650	kJ/kmol
HW	-9.99298	?	-9.99298	0.997042	kJ/kg
AIR.1_MOLF	215000.	21500.0	186103.	4967.11	Nm ³ /h
MASSF	66.5413	?	66.5413	1.77600	kg/s
HFLOW	-664.945	?	-664.945	67.8622	kW
AIR.1_MFO2	21.0000	CST	21.0000	0.00000	%

TAGREP (General)

Tags (type: Input).

0

0: (none)

No Tags report

1

1: Tags definitions

Table with the definitions of all the tags:

TAGS DEFINITION							
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit	From file
AIR_1_MFO2	STREAM	AIR.1	MFO2	21.000	CST	%	
AIR_1_MOLF	STREAM	AIR.1	MOLF	0.21500E+06	-10.00	Nm3/h	
AIR_1_P	STREAM	AIR.1	P	1.0130	-2.000	bar	
AIR_1_T	STREAM	AIR.1	T	15.000	1.00	C	
AIR_2_P	STREAM	AIR.2	P	20.000	-2.000	bar	
AIR_2_T	STREAM	AIR.2	T	500.00	2.00	C	
C-1_EFFIC	UNIT	C-1	EFFIC	77.000	-5.000	%	
FUME_1_T	STREAM	FUME.1	T		OFF	C	
FUME_2_MFO2	STREAM	FUME.2	MFO2	14.000	-2.000	%	
FUME_2_P	STREAM	FUME.2	P	1.0200	-2.000	bar	
FUME_2_T	STREAM	FUME.2	T	600.00	2.00	C	

2

2: Tags values

Table with the measured and validated values of all the tags:

TAGS IN THE MODEL							
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit	Comment
AIR_1_MFO2	21.000	CST	21.000			%	
AIR_1_MOLF	0.21500E+06	10.0 %	0.18610E+06	2.67 %	1.81	Nm3/h	
AIR_1_P	1.0130	2.00 %	1.0117	1.97 %	0.00	bar	
AIR_1_T	15.000	1.00	15.049	0.994	0.00	C	
AIR_2_P	20.000	2.00 %	20.016	1.40 %	0.00	bar	
AIR_2_T	500.00	2.00	499.93	1.99	0.00	C	
C-1_EFFIC	77.000	5.00 %	75.681	1.33 %	0.12	%	
FUME_1_T		OFF	1202.5	0.403 %		C	
FUME_2_MFO2	14.000	2.00 %	14.058	0.434 %	0.04	%	
FUME_2_P	1.0200	2.00 %	1.0197	1.99 %	0.00	bar	
FUME_2_T	600.00	2.00	599.95	1.99	0.00	C	

3

3: Full Report

Both tables are printed

4

4: VDI-2048

Same as 3, but the following values are printed: measured value, correction, validated value, relative correction penalty and physical units:

TAGS IN THE MODEL							
Tag name	Mea. Val.	Correction	Rec. Val.	Rel.Corr.	Penalty	Unit	Comment
%_ECOWO			24.636			%	
%_REHEAT			12.779			%	
%_SUPHEAT			31.588			%	
%_VAPO			29.058			%	
1_P	1.0130	0.0000	1.0130	0.000 %	0.000	bar	
1_T	25.000	0.0000	25.000	0.000 %	0.000	C	
2_P	1.0130	0.0000	1.0130	0.000 %	0.000	bar	
2_T	25.000	0.0000	25.000	0.000 %	0.000	C	
AIR_0_HFLOWTOT			-0.69693			NW	
AIR_0_HUMIDITY	30.001	0.59159E-02	30.007	0.1971E-01 %	0.000	%	
AIR_0_MASSF			305.20			kg/s	
AIR_0_MGO2	21.000	0.0000	21.000	0.000 %	0.000	%	
AIR_0_MOLF	0.86778E+06	-12574.	0.85521E+06	-1.470 %	0.021	Nm3/h	

5

5: No r5v,r6v

No Tags report and no r5v and r6v files

REACREP (General)

Reactions (type: Input).

0 0: (none)

No report on reactions

1 1: Stoichiometry

Table with the stoichiometry of the reactions:

REACTIONS			
SYNT			
reaction comment			
N2 + 3 H2 <-> 2 NH3			
reaction definition			
3.00 H2 + 1.00 N2 <=> 2.00 NH3			
REACTION DATA			
KA = -1	KB = -726.850	ETYPE = 1	K0 = -1
Act. En. = -1	VTYP = 1	DHR = -1	TREF = -726.850

2 2: Full

Table with the stoichiometry and kinetic data of the reactions

REACTIONS			
SYNT			
reaction comment			
N2 + 3 H2 <-> 2 NH3			
reaction definition			
3.00 H2 + 1.00 N2 <=> 2.00 NH3			
REACTION DATA			
KA = -1	KB = -726.850	ETYPE = 1	K0 = -1
Act. En. = -1	VTYP = 1	DHR = -1	TREF = -726.850

THERREP (General)

Thermods (type: Input).

0 0: (none)

No report on THERMODS

1 1: Yes

Table with the definitions of the THERMODS (methods, compounds and binary parameters):

THERMODYNAMIC MODELS		
STEAM		
CONTAINS: H2O,		
THERMODYNAMIC METHODS USED		
Fugacity	IAPWS-IF97	IAPWS-IF97
Enthalpy	IAPWS-IF97	IAPWS-IF97
Volume	IAPWS-IF97	IAPWS-IF97
Conductibility	IAPWS-IF97	IAPWS-IF97
Viscosity	IAPWS-IF97	IAPWS-IF97
Vol. Pres. drop	IAPWS-IF97	IAPWS-IF97
BINARY PARAMETERS DEFINED		
No binary parameters defined		

COMPREP (General)

Compounds (type: Input).

- 0 0: (none) No report on COMPOUNDS
- 1 1: One column report Table with the parameters of the COMPOUNDS in one column:

C1	
++	
C 1 H 4	
COMPOUND Comment: METHANE	
HOF	-74835.0000
GOF	-50785.2656
MOIW	16.0429993
TB	111.657989
TC	190.549988
PC	46.0420685
VC	0.989999771E-01
OMEGA	0.719999995E-02
CPLA	82.5483856
CPLB	-0.693272591
CPLC	0.483209919E-02
CPLD	-0.720117441E-05
TMIN (FV)	81.6999970
TMAX (FV)	186.787842
PVAPA	13.0173359
PVAPB	-1192.28540
...	

VARREP (General)

Variables/Constants (type: Input).

- 0 0: (none) No Variables/Constants report
- 1 1: Unmeasured Variables Table with the values and a posteriori accuracy of unmeasured variables

UNMEASURED VARIABLES IN THE MODEL						
Object	Owner	Var. name	Rec. val.	Rec. acc.	Rec. acc. (%)	Unit
STREAM	AIR.1	H	-288.301	28.8	9.98 %	kJ/kmol
RATES	AIR.1	N2	0.147021E+006	0.392E+04	2.67 %	Nm3/h
RATES	AIR.1	O2	39081.5	0.104E+04	2.67 %	Nm3/h
STREAM	AIR.2	H	14496.4	63.6	0.439 %	kJ/kmol
RATES	FUME.1	CO2	6035.58	114.	1.90 %	Nm3/h
STREAM	FUME.1	H	39228.3	226.	0.576 %	kJ/kmol
RATES	FUME.1	H2O	12071.2	229.	1.90 %	Nm3/h
RATES	FUME.1	N2	0.147021E+006	0.392E+04	2.67 %	Nm3/h
RATES	FUME.1	O2	27010.4	821.	3.04 %	Nm3/h
STREAM	FUME.1	P	19.9156	0.280	1.41 %	bar
STREAM	FUME.1	T	1202.53	5.95	0.403 %	C
STREAM	FUME.2	H	18149.0	68.2	0.376 %	kJ/kmol
UNIT	G-1	TOTPOWOU	50194.7	917.	1.83 %	kW
RATES	NG	C1	6035.58	114.	1.90 %	Nm3/h
STREAM	NG	H	-352.444	35.0	9.92 %	kJ/kmol
RATES	NG	MFC1	100	0.00	0.00 %	%
STREAM	W-1	POWER	34100.0	922.	2.70 %	kW
STREAM	W-2	POWER	50194.7	917.	1.83 %	kW

- 2 2: Constants Table with the values of all the constants

CONSTANT VARIABLES IN THE MODEL						
Object	Owner	Var. name	Value	Unit	Tag	
RATES	AIR.1	MFO2	21	%		AIR.1_MFO2
RATES	FUME.1	SO2	0	Nm3/h		
RATES	NG	C2	0	Nm3/h		
RATES	NG	C3	0	Nm3/h		
RATES	NG	CO2	0	Nm3/h		
RATES	NG	H2	0	Nm3/h		
RATES	NG	H2S	0	Nm3/h		
RATES	NG	I-C4	0	Nm3/h		
RATES	NG	MFC2	0	%		NG_MFC2
RATES	NG	MFC3	0	%		NG_MFC3
RATES	NG	MFCO2	0	%		NG_MFCO2
RATES	NG	MFH2	0	%		NG_MFH2
RATES	NG	MFH2S	0	%		NG_MFH2S
RATES	NG	MFI-C4	0	%		NG_MFI-C4
RATES	NG	MFN-C4	0	%		NG_MFN-C4
RATES	NG	MFN-C5	0	%		NG_MFN-C5
RATES	NG	MFN2	0	%		NG_MFN2
RATES	NG	N-C4	0	Nm3/h		
RATES	NG	N-C5	0	Nm3/h		
RATES	NG	N2	0	Nm3/h		

- 3 3: Unmeasured Variables & Both tables are printed
Constants
- 4 4: idem 3 + Hidden variables Same as 3, but intermediate variables are also printed

BNDREP (General)

Variable Bounds (type: Input).

0 0: (none)

No report on the bounds

1 1: Active Bounds

Table with all the active (solver) bounds defined on the variables:

BOUNDS									
Object	Owner	Var. name	Low Bound	Value	High Bound	Rec. acc.	L/H	Unit	Tag
UNIT	C-1	EFFIC	76	76.0002	100	0.00	SLC	%	C-1_EFFIC

2 2: Bounds on all the variables

Table with all the (solver) bounds defined on the variables:

***** * BOUNDS ON VARIABLES * *****									
SLC/SHC : Low/High Bound Generating a Constraint SLI/SHI : Low/High Bound Detected by Solver									
OBJECT	OWNER	VAR.NAME	LOW BOUND	REC.VAL.	UP BOUND	REC.ACC.	L/H	P.U.	
=====									
STREAM	AIR.1	H	-0.100E+11	-285.697	0.100E+11	27.1		kJ/kmol	
RATES	AIR.1	MF02	0.00	21.0000	100.		CST	%	AIR.1_MF02
RATES	AIR.1	MOLF	0.00	190115.	0.807E+10	0.462E+04		Nm3/h	AIR.1_MOLF
RATES	AIR.1	N2	0.00	1.86132	0.100E+06	0.453E-01		kmol/s	
RATES	AIR.1	O2	0.00	0.494782	0.100E+06	0.120E-01		kmol/s	
STREAM	AIR.1	P	0.100E-01	1.00678	0.100E+04	0.156E-01		bar	AIR.1_P
STREAM	AIR.1	T	-43.2	15.1392	0.373E+04	0.938		C	AIR.1_T
STREAM	AIR.2	H	-0.100E+11	14494.0	0.100E+11	61.3		kJ/kmol	
STREAM	AIR.2	P	0.100E-01	20.0650	0.100E+04	0.309		bar	AIR.2_P
STREAM	AIR.2	T	-43.2	499.856	0.373E+04	1.92		C	AIR.2_T
UNIT	BURNER	DP	0.00	0.999991E-01	0.100E+04	0.100E-01		bar	BURNER_DP
UNIT	COMP	EFFIC	76.0	76.0007	100.	0.00	SLC	%	COMP_EFFIC
RATES	FUME.1	CO2	0.00	0.759138E-01	0.100E+06	0.132E-02		kmol/s	
STREAM	FUME.1	H	-0.100E+11	39069.7	0.100E+11	206.		kJ/kmol	
...									

COBAREP (General)

Compounds Balance (type: Input).

0 0: (none)

No compounds balance

1 1: Molar basis

Table with the partial molar rates of all compounds in the global inlet and outlet streams of the flow-sheet:

OVERALL MATERIAL BALANCES (KMOL/s)						
IN	O2	N2	CO2	H2O	SO2	H2
AIR.1	0.48435	1.82208				
NG	0.48435					
TOTAL	0.48435	1.82208	0	0	0	0
OUT	O2	N2	CO2	H2O	SO2	H2
FUME.2	0.33475	1.82208	0.748009E-001	0.14960		
TOTAL	0.33475	1.82208	0.748009E-001	0.14960	0	0

2 2: Mass basis

Table with the partial mass rates of all compounds in the global inlet and outlet streams of the flow-sheet:

OVERALL MATERIAL BALANCES (kg/s)						
IN	O2	N2	CO2	H2O	SO2	H2
AIR.1	55795.1	0.183754E+006				
NG	55795.1					
TOTAL	55795.1	0.183754E+006	0	0	0	0
OUT	O2	N2	CO2	H2O	SO2	H2
FUME.2	38561.6	0.183754E+006	11851.1	9702.49		
TOTAL	38561.6	0.183754E+006	11851.1	9702.49	0	0

3 3: Molar and Mass basis

Both tables are printed

INDREP (General)

Indexes at the bottom (type: Input).

These indexes give an easy access to the units and streams information.

- 0 0: (none) No index
- 1 1: Units Index Alphabetically sorted table with the units, number of equations, and mass balance:

UNITS SUMMARY												
Name	Type	Streams		Mode	Equations					Mass balance		
		In	Out		Mass	Heat	LV	P	Other	Mea. val.	Rec. val.	
C-1	COMVAL	2	1	0	0	1	0	0	1	10.332	0.0000	kg/s
G-1	BBXVAL	1	2	0	0	2	0	0	0	0.0000	0.0000	kg/s
R-1	BATVAL	2	1	0	4	1	0	2	0	-0.70734E-02	0.57073E-06	kg/s
T-1	TURVAL	1	2	0	0	1	0	0	1	0.0000	0.0000	kg/s

- 2 2: Streams Index Alphabetically sorted list of streams:

STREAMS		
Stream name	Origin	Destination
AIR.1		C-1
AIR.2	C-1	R-1
FUME.1	R-1	T-1
FUME.2	T-1	
NG		R-1
POWER	G-1	
W-1	G-1	C-1
W-2	T-1	G-1

- 3 3: Units and Streams Indexes Both indexes are printed

IMAGE (General)

Images (type: Input).

- 0 0: No images in HTML No images are included in the html, which insures full portability of the report to other computers
- 1 1: Images in HTML Some images are include in the html. These images are located in belsim/dat directory and will only be correctly shown if present on the computer where the report is browsed
- 2 2: Local Dir Images in HTML Some images are include in the html and are copied in the local directory. When moving or copying the html report you should also move or copy the corresponding GIF files

ORDVAREP (General)

Ordering of variables in variables table (type: Input).

- 0 0: Alphabetical order Variables are sorted alphabetically
- 1 1: decreasing relative rec.acc.Variables are sorted by decreasing relative uncertainty ordering

ZONEBAL (General)

Balances by zone (type: Input).

0	0: Disabled	Product balances are generated globally
1	1: Enabled	Product balances are generated globally and by zone

When generating balances by zone, product balance tags are generated for each zone (starting by the name of the zone as defined in the parent PFD).

Example:

A1_IMPORT_CRUDE_MASSF will be the total imported amount (in mass) of crude into zone A1.

ORDTAREP (General)

Ordering of tags in Tags table (type: Input)

0	0: Alphabetical order	Tags are sorted alphabetically
1	1: Decreasing penalties ordering	Tags are sorted by decreasing penalty

SECTION 39: RKIVAL : REACTOR WITH KINETICS

PURPOSE

The **RKIVAL** unit allows you to model reactors and takes the atomic balances, energy balance and pressure drop into account.

The chemical reactions must be defined and you can give the kinetic equations of these reactions.

SYMBOLS

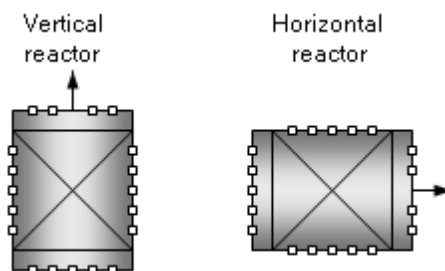


FIGURE 58 - SYMBOLS FOR UNIT RKIVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 2
- Type: 1 material (mandatory) and 1 thermal (optional)

OUTLET STREAMS

- Maximum number: 2
- Type: 1 material as the first outlet (mandatory) and 1 thermal (optional)

THERMOD

All chemical compounds involved in the chemical reactions must be present in the outlet THERMOD.

REACTIONS

- Maximum number: 30

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account

NAME	TAB	TYPE	DESCRIPTION
FLOWMODE	General	Input	Selection of the reactor's type
REACTYPE	General	Input	Selection of the reactions type
MODEDP	General	Input	Pressure equation
FLEXMODE	General	Input	Flag to specify the flex utilization
DIREAC	Parameters	Measurable	Diameter of the reactor
HREAC	Parameters	Measurable	Height of the reactor
ROCAT	Parameters	Measurable	Specific mass of the catalyst
DP	Parameters	Measurable	Pressure drop
DPX	Parameters	Input	Nominal pressure drop
MASSF	Parameters	Input	Nominal flow rate
CFDP	Parameters	Measurable	DP correction factor
REFDP	Parameters	Input	Reference stream for pressure drop
PHASE	Parameters	Input	Deviation from 0 or 1 on vapor fraction
EXTENTi	Reactions	Measurable	Extents of the reactions

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	All equations	All equations (including kinetics) are taken into account
1	Mass Balance only	Only the mass balances are taken into account
2	Mass and Energy balance	Atomic and energy balances are taken into account

FLOWMODE (General)

Selection of the reactor's type (type: Input).

1	Perfectly Mixed Flow	Reactor is considered as perfectly mixed
2	Plug Flow	Plug Flow

REACTYPE (General)

Selection of the reactions type (type: Input).

1	Homogeneous reactions	Homogeneous reactions
2	Heterogeneous reactions	Heterogeneous reactions

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * (MASSF / MASSFX)^2$$

where:

- CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%)
- DPX: the nominal pressure drop
- MASSFX: the nominal flow rate
- MASSF: the actual flow rate

FLEXMODE (General)

Flag to specify the flex utilization (type: Input).

1	With Flex	The kinetics defined in the Flex code are used
2	Without Flex	The kinetics defined in the Flex code are not used

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

DIREAC (Parameters)

Diameter of the reactor (type: Measurable).

HREAC (Parameters)

Height of the reactor (type: Measurable).

ROCATA (Parameters)

Specific mass of the catalyst (type: Measurable).

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when MODEDP = 2.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when MODEDP = 2.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when MODEDP = 2.

PHASE (Parameters)

Deviation from 0 or 1 on vapor fraction (type: Input).

In the interval $[0, \text{PHASE}]$, stream is considered as liquid.

In the interval $[1 - \text{PHASE}, 1]$, stream is considered as vapor.

In the interval $[\text{PHASE}, 1 - \text{PHASE}]$, there is no calculation and program stops.

EXTENT_i (Reactions)

Extents of the reactions (type: Measurable).

The EXTENT of a reaction is the amount of reference product (with stoichiometric coefficient equal to 1) produced in the associated reaction.

For a particular component, the amount produced in a given reaction is the result of the multiplication of the EXTENT of the reaction by the stoichiometric coefficient of the component (thus value is negative for reagents).

EXAMPLE : NH₃ SYNTHESIS REACTOR

Attention: see model rkival.bls in the directory "examples\rkival".

SECTION 40: RKPVAL : EQUILIBRATED REACTIONS

PURPOSE

The **RKPVAL** unit allows you to model reactors where some of the reactions are equilibrated and takes the atomic balances, energy balance and pressure drop into account.

SYMBOLS

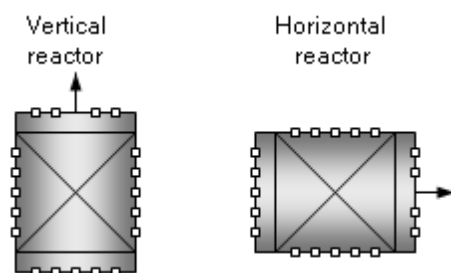


FIGURE 59 - SYMBOLS FOR UNIT RKPVAL

CONNECTION RULES

INLET STREAMS

- Maximum number: 10
- Type: material (at least one), thermal, mechanical

OUTLET STREAMS

- Maximum number: 10
- Type: one material as the first outlet (as shown on the symbols), thermal, mechanical

THERMOD

All chemical compounds involved in the chemical reactions must be present in the outlet THERMOD.

REACTIONS

- Maximum number: 30

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account

NAME	TAB	TYPE	DESCRIPTION
MODEDP	General	Input	Pressure drop balance
DP	Parameters	Measurable	Pressure drop
DPX	Parameters	Input	Nominal pressure drop
MASSFx	Parameters	Input	Nominal flow rate
CFDP	Parameters	Measurable	DP correction factor
REFDP	Parameters	Input	Reference stream for pressure drop
EXTENTi	Reactions	Measurable	Extents of the reactions
DTEQi	Reactions	Measurable	Temperature deviation from equilibrium
EQi	Reactions	Input	Indicates whether the reaction is equilibrated

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	Atomic and Energy Balance	Atomic and energy balances are taken into account
1	Atomic Balance only	Only atomic balances are taken into account

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form: $DP = CFDP * DPX * (MASSF / MASSFX)^2$ where: ■ CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%) ■ DPX: the nominal pressure drop ■ MASSFX: the nominal flow rate ■ MASSF: the actual flow rate

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the reference inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if $MODEDP = 0$.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

MASSFX (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

EXTENTi (Reactions)

Extents of the reactions (type: Measurable).

The EXTENT of a reaction is the amount of reference product (with stoichiometric coefficient equal to 1) produced in the associated reaction.

For a particular component, the amount produced in a given reaction is the result of the multiplication of the EXTENT of the reaction by the stoichiometric coefficient of the component (thus value is negative for reagents).

DTEQi (Reactions)

Temperature deviation from equilibrium (type: Measurable).

The DTEQi parameter allows you to account for a deviation from equilibrium by calculating the equilibrium constant at a temperature different ($T + DTEQi$) from the outlet stream temperature T .

Attention: When $DTEQ > 100$ C, no equilibrium constraint is generated. This has the same effect as setting the parameter EQi to 0.

EQi (Reactions)

Indicates whether the reaction is equilibrated (type: Input).

0	Conversion	The reaction is not equilibrated
1	Equilibrium	The reaction is equilibrated

EXAMPLE : NH3 SYNTHESIS REACTOR

Attention: see model `rkpval.bls` in the directory "examples\rkpval".

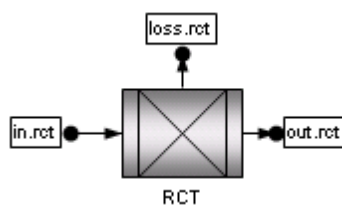


FIGURE 60 - AN EXAMPLE OF RKPVAL TYPE UNIT

This example deals with the same chemical reactor for the synthesis of NH₃ as in the example for RCTVAL. We use here the RKPVAL to determine the distance from the equilibrium composition. This difference expressed in C can be considered as an indication of catalyst ageing.

There is 1 UNIT:

- RCT
 - type=RKPVAL
 - MODE='Atomic and Energy Balance'
 - MODEDP='DP Variable'
 - EQ1='Equilibrium' (reaction page)

The reaction SYNT must be defined as: $3 \text{ H}_2 + 2 \text{ N}_2 \rightleftharpoons 2 \text{ NH}_3$

3 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
IN.RCT	RCMOD	to RCT
OUT.RCT	RCMOD	from RCT
LOSS.RCT	RCMOD	from RCT

The THERMOD RCMOD can be derived from the THERMOD AIR available in Empty.bls and contains: H₂, N₂, NH₃ and Ar.

The list of measurements and the results are given below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
IN.RCT_MFAR	STREAM	IN.RCT	MFAR	5.7000	-5.000	%
IN.RCT_MFH2	STREAM	IN.RCT	MFH2	70.000	-5.000	%
IN.RCT_MFN2	STREAM	IN.RCT	MFN2	21.000	-5.000	%
IN.RCT_MFNH3	STREAM	IN.RCT	MFNH3	3.3000	-5.000	%
IN.RCT_MOLF	STREAM	IN.RCT	MOLF	0.78000E+06	-5.000	Nm3/h
IN.RCT_P	STREAM	IN.RCT	P	200.00	-2.000	bar
IN.RCT_T	STREAM	IN.RCT	T	282.00	1.00	C
OUT.RCT_MFAR	STREAM	OUT.RCT	MFAR	6.0000	-5.000	%
OUT.RCT_MFH2	STREAM	OUT.RCT	MFH2	61.000	-5.000	%
OUT.RCT_MFN2	STREAM	OUT.RCT	MFN2	17.000	-5.000	%
OUT.RCT_MFNH3	STREAM	OUT.RCT	MFNH3	16.000	-5.000	%
OUT.RCT_P	STREAM	OUT.RCT	P	198.00	-2.000	bar
OUT.RCT_T	STREAM	OUT.RCT	T	470.00	2.00	C
RCT_DP	UNIT	RCT	DP	1.5000	-10.00	bar
RCT_DTEQ1	UNIT	RCT	DTEQ1	10.000	0.100E+04	K

TABLE 37 - MEASUREMENT LIST FOR RKPVAL EXAMPLE

Number of tags	15					
Number of equations	18					
Number of unmeasured variables	12					
Number of measured variables	15					
Number of variables considered constant	0					
Obvious number of redundancies	6					
Total number of redundancies	6					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	0.95649					
Chi-square test value	12.6000					
Sum of squares of residues	0.638906E-014					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
IN.RCT_MFAR	5.7000	5.00 %	5.5114	3.55 %	0.44	%
IN.RCT_MFH2	70.000	5.00 %	70.449	0.902 %	0.02	%
IN.RCT_MFN2	21.000	5.00 %	20.738	2.89 %	0.06	%
IN.RCT_MFNH3	3.3000	5.00 %	3.3023	4.99 %	0.00	%
IN.RCT_MOLF	0.78000E+06	5.00 %	0.78000E+06	5.00 %	0.00	Nm3/h
IN.RCT_P	200.00	2.00 %	199.75	1.41 %	0.00	bar
IN.RCT_T	282.00	1.00	282.00	1.00	0.00	C
OUT.RCT_MFAR	6.0000	5.00 %	6.1863	3.53 %	0.39	%
OUT.RCT_MFH2	61.000	5.00 %	60.707	1.59 %	0.01	%
OUT.RCT_MFN2	17.000	5.00 %	17.154	3.92 %	0.03	%
OUT.RCT_MFNH3	16.000	5.00 %	15.953	4.80 %	0.00	%
OUT.RCT_P	198.00	2.00 %	198.25	1.42 %	0.00	bar
OUT.RCT_T	470.00	2.00	470.00	2.00	0.00	C
RCT_DP	1.5000	10.0 %	1.5004	9.99 %	0.00	bar
RCT_DTEO1	10.000	0.100E+04	25.534	6.72	0.00	K

TABLE 38 - RESULTS OF RKPVAL EXAMPLE

SECTION 41: RLVVAL : REACTOR IN L/V EQUILIBRIUM

PURPOSE

UNIT RLVVAL is a unit modeling a reactor where outlet streams are at liquid-vapor equilibrium.

SYMBOLS

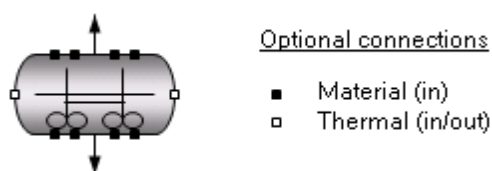


FIGURE 61 - SYMBOLS FOR UNIT RLVVAL

CONNECTION RULES

INLET STREAMS

- Maximum number:
 - material: 8
 - thermal: 2

OUTLET STREAMS

- Maximum number:
 - material:
 - 1 stream of state V as vapor outlet (upper stream)
 - 1 stream of state L as liquid outlet (lower stream)
 - thermal: 2

THERMODS

Both outlet streams must use the same THERMOD.

All chemical compounds involved in the chemical reactions must be present in the outlet THERMOD.

REACTIONS

- Maximum number: 30

The state of the outlet material streams can also be LV, LLV and WLV provided that there is more than one component in the stream.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
DP	Parameters	Measurable	Pressure drop
REFDP	Parameters	Input	Reference stream for pressure drop
EXTENTi	Reactions	Measurable	Extents of the reactions

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	All equations	Atomic and energy balances are taken into account
1	Atomic Balance only	Only atomic balances are taken into account
2	Atomic and Energy Balance	Atomic and energy balances are taken into account
3	Mass Balance and Saturation	Atomic balances and liquid-vapor equilibrium relationships are taken into account

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the reference inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if MODEDP = 0.

EXTENT_i (Reactions)

Extents of the reactions (type: Measurable).

The EXTENT of a reaction is the amount of reference product (with stoichiometric coefficient equal to 1) produced in the associated reaction.

For a particular component, the amount produced in a given reaction is the result of the multiplication of the EXTENT of the reaction by the stoichiometric coefficient of the component (thus value is negative for reagents).

SECTION 42: SATVAL : SATURATION UNIT

PURPOSE

This type of UNIT allows you to set the degree of saturation of a streams. It is possible to fix any vapor fraction (between 0 and 1) with a possible temperature departure from the temperature corresponding to a specified vapor fraction. Heat can be extracted or provided to this unit.

SYMBOLS



FIGURE 62 - SYMBOLS FOR UNIT SATVAL

CONNECTION RULES

INLET STREAMS

- 1 material
- 1 thermal or mechanical (optional)

OUTLET STREAMS

- 1 material
- 1 thermal or mechanical (optional)

The outlet Stream can be set in state L, V or LV. It is however recommended to use the L state when the outlet stream that is specified at the bubble point, and the V state when the outlet stream is specified at the dew point.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
DP	Parameters	Measurable	Pressure drop
DPX	Parameters	Input	Nominal pressure drop
MASSFx	Parameters	Input	Nominal flow rate

NAME	TAB	TYPE	DESCRIPTION
CFDP	Parameters	Measurable	DP correction factor
VFRAC	Parameters	Input	Saturation state
DT	Parameters	Input	Level of superheating or subcooling

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	All equations	Mass, energy balances and saturation constraints are taken into account
1	Mass Balance only	Only mass balances are taken into account
2	Mass and Energy Balance	Mass and energy balances are taken into account
3	Mass Balance and Saturation	Mass and energy balances are taken into account

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated
2	DP=F(MASSF)	A pressure drop equation will be generated and the pressure drop will be a function of the mass flow rate according to an equation having the following form:

$$DP = CFDP * DPX * (MASSF / MASSFX)^2$$

where:

- CFDP: a measurable parameter that accounts for the deviation from the calculated value (typically measured at 1 with a precision of 20 to 30%)
- DPX: the nominal pressure drop
- MASSFX: the nominal flow rate
- MASSF: the actual flow rate

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the reference inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if $MODEDP = 0$.

DPX (Parameters)

Nominal pressure drop (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

MASSFx (Parameters)

Nominal flow rate (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

CFDP (Parameters)

DP correction factor (type: Input).

Used in the pressure drop equation when $MODEDP = 2$.

VFRAC (Parameters)

Saturation state (0..1) (type: Input).

This parameter defines the ratio of vapor in the mixture. The value of this parameter has to be set in the range from 0 to 1: if the value is out of this range, it will be corrected to the closer value.

This parameter is always a constant.

DT (Parameters)

Level of superheating or subcooling (type: Input).

This parameter defines the temperature difference between the temperature of the thermodynamic state defined from the vapor fraction parameter and the temperature of the real thermodynamic state of the STREAM.

With this parameter, it is possible to define how much a liquid is subcooled or a vapor is superheated.

This parameter is always a constant.

EXAMPLE : SUB-COOLED WATER OF FLOW

Attention: see model satval.bls in the directory "examples\satval".

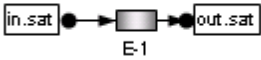


FIGURE 63 - SATVAL EXAMPLE

There is 1 UNIT:

- E-1:
 - type=SATVAL
 - MODE='All equations'
 - MODEDP='DP variable'
 - DT= -5 K

2 STREAMS must be defined:

STREAMS THERMOD		CONNECTIONS
IN.SAT	STEAM	to E-1
OUT.SAT	STEAM	from E-1

The Thermod STEAM is defined in the file Empty.bls.

The list of measurements are given in tables below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
IN.SAT_MASSF	STREAM	IN.SAT	MASSF	500.00	-1.000	kg/s
IN.SAT_P	STREAM	IN.SAT	P	50.000	1.00	bar
IN.SAT_T	STREAM	IN.SAT	T	260.00	3.00	C
OUT.SAT_T	STREAM	OUT.SAT	T	257.00	3.00	C

TABLE 39 - MEASUREMENTS FOR SATVAL EXAMPLE

Number of tags	4					
Number of equations	7					
Number of unmeasured variables	5					
Number of measured variables	4					
Number of variables considered constant	1					
Obvious number of redundancies	2					
Total number of redundancies	2					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	0.52789					
Chi-square test value	5.99000					
Sum of squares of residues	0.403042E-006					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
IN.SAT_MASSF	500.00	1.00 %	500.00	1.00 %	0.00	kg/s
IN.SAT_P	50.000	1.00	49.915	0.862	0.01	bar
IN.SAT_T	260.00	3.00	258.81	1.08	0.16	C
OUT.SAT_T	257.00	3.00	258.81	1.08	0.36	C

TABLE 40 - RESULTS OF SATVAL EXAMPLE

SECTION 43: SPLVAL : SPLITTER

PURPOSE

This UNIT is used to consider heat and mass balances of a splitter where the outlet STREAMS have the same composition.

SYMBOLS

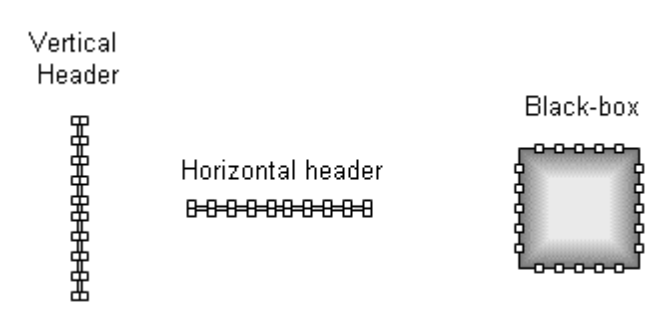


FIGURE 64 - SYMBOLS FOR UNIT SPLVAL

CONNECTION RULES

INLET STREAMS

- 1 material

OUTLET STREAMS

- Maximum number: 10
- Type: material

THERMOD

Inlet and outlet streams must use the same THERMOD.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
MODETOE	General	Input	T equalities (mass balance only)

NAME	TAB	TYPE	DESCRIPTION
Q			
DP	Parameters	Measurable	Pressure drop
FRACi	Parameters	Measurable	Split Fractions
FLOWFILT	General	Input	Automatic flow rate filtering

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Mass and Energy Balance	Mass and energy balances are taken into account
1	Mass Balance only	Only mass balances are taken into account

MODEDP (Parameters)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

MODETOEQ (General)

Temperature equalities (type: Input).

0	None	no temperature equality equation will be generated
1	between all outlet stream	all outlet streams will be set at the same temperature
2	between first inlet and outlet stream	all outlet streams and the first inlet stream will be set at the same temperature

Temperature equalities are only added when the unit is calculated in mass balance only. When the energy balance is activated, these additional equations will never be generated.

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$DP = P_{in} - P_{out}$

This equation is not generated if MODEDP = 0.

where:

- P_{in} : the pressure of the inlet stream
- P_{out} : the pressure of the outlet stream

FRACi (Parameters)

Split Fractions (type: Measurable).

Each FRACi parameter represents the fraction of the inlet flow going in the corresponding outlet stream.

FLOWFILT (Parameters)

Automatic flow rate filtering (type: Input).

0	Disabled	The automatic flow rate filtering is not activated
1	Enabled	The automatic flow rate filtering is activated

When the inlet flow rate of a splitter is measured as well as all **but one** of its outlet streams, there is a risk that the unmeasured outlet stream hits its lower bound.

This in particular the case on products routing in production accounting applications where the main production flow rate is measured and the product routing is specified though manual inputs on the flows that are redirected from the normal route (on-spec product) to slop tanks or recirculation lines.

The automatic filtering (when enabled) creates an estimate on the unmeasured flow rate based on the measured inlet and outlet flow rates. Should the estimated flow rate fall below a given cutoff value, the unmeasured stream will be set off.

SECTION 44: TNKVAL : STORAGE TANK

PURPOSE

This type of UNIT allows you to consider the material balance of a storage tank. Different modes can be selected to model the tank: mass, volume or level basis, use of strapping tables, tracking of composition change over the time window. In any case, the heat balance of the tank is never taken into account.

Typically, storage tanks models are most often used in daily production balance models. In such a case, the recommended approach is to use as initial inventory (measured as constants) the validated inventory of the preceding period (usually day).

Attention: Unaccounted losses (or surpluses) will then be evidenced by a progressive drift between the validated and measured inventory of the tank. From time to time (for example at the end of a monthly period), validated tank inventories can be reset to measured inventory by enabling an imbalance on the tank inventory (parameter IMBMODE).

The inventory change can be expressed on a mass basis or a volume basis

In case of a mass basis, you can measure

- directly the mass inventory change (parameter DMASS),
- the initial and final mass inventories (parameters MASS1 and MASS2)
- or the initial and final tank levels, using a strapping table to convert levels in mass inventories

When dealing with volume basis, you will measure:

- the initial and final volume inventories (parameters VOL1 and VOL2)
- or the initial and final tank levels and use a strapping table to convert them volume inventories
- at least the density of the initial contents of the tank

In such cases, volumes need to be converted into mass terms to correctly calculate the material balance. The density will be calculated either on basis of the composition of the tank contents (COLDDENS approach) or on basis of a so-called conventional density which will be a measured property independent of the tank composition (CONVDENS approach).

Several approaches are also possible to track the composition of the tank inventories:

- assuming that the composition of the tank and of the connected streams are not related
- assuming that the composition of the tank remains constant over the validation time window
- assuming that the composition of the tank can change over time

In the latter case, the initial composition of the tank needs to be measured. As for the inventory, the

usual approach is to use as initial composition (measured as constants) the validated composition of the preceding period (usually day).

There is also the possibility to specify that the tank is used only in charge or rundown operation (no simultaneous input and output) during the validation period of time, which avoids the indetermination introduced by tanks used in parallel.

SYMBOLS



FIGURE 65 - SYMBOLS FOR UNIT TNKVAL

CONNECTION RULES

INLET STREAMS

- 1 optional material STREAM

OUTLET STREAMS

- 1 optional material STREAM

At least one STREAM must be connected to the unit (inlet or outlet).

THERMOD

Inlet and outlet streams must use the same THERMOD.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
VOLMODE	General	Input	Volume mode
COMPMODE	General	Input	Compound balance mode
NTABLE	General	Input	Strapping Table
MODECURV	General	Input	Interpolation method

NAME	TAB	TYPE	DESCRIPTION
OPTMODE	General	Input	Validation method
INITDENS	General	Input	Density initialization mode
IMBMODE	General	Input	Imbalance mode
DMASS	Parameters	Measurable	Accumulation (Delta Mass)
DTIME	Parameters	Input	Time step (Delta Time)
WIMBAL	Parameters	Measurable	Weight Imbalance
VIMBAL	Parameters	Measurable	Volume Imbalance
MASS1,2	Others	Measurable	Mass in tank at time t1 and t2
VOL1,2	Others	Measurable	Volume in tank at time t1 and t2
DENS1,2	Others	Measurable	Density in tank at time t1 and t2
LEVEL1,2	Others	Measurable	Level at time t1 and t2
LVLFRA1,2	Others	Measurable	Level Fraction at time t1 and t2

Additional measurements available when dealing with variable composition:

LG1,2	Compound dry liquid volume fraction at time t1 and t2
MG1,2	Compound dry molar fraction at time t1 and t2
WG1,2	Compound dry weight fraction at time t1 and t2
LF1,2	Compound liquid volume fraction at time t1 and t2
CC1,2	Compound mass concentration at time t1 and t2
MF1,2	Compound molar fraction at time t1 and t2
WF1,2	Compound weight fraction at time t1 and t2
WQ1,2	Compound weight quantity at time t1 and t2
V _i	Compound volume imbalance during the validation period
W _i	Compound weight imbalance during the validation period

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

1	Mass Balance	Only the mass balance is considered.
10	No simultaneous Input and Output	If the measured initial and final inventories indicate that the tank is filling up, then the outlet stream is set OFF, and vice versa. The measured initial and final inventories that are checked are: 1. MASS1 and MASS2 2. DMASS 3. VOL1 and VOL2 4. LEVEL1 and LEVEL2 5. LVLFRA1 and LVLFRA2

VOLMODE (General)

Volume mode (type: Input).

0	None	Volumes are not taken into account, only masses are considered
1	Volume Basis	Volumes are taken into account

The density and volume variables are taken into account, assuming that the tank is at 15°C. The densities used by the unit must be linked to the streams densities by setting the unit comment to either COLDDENS (density calculated from composition) or CONVDENS (density based on conventional density as measured on the stream).

COMPMODE (General)

Compound balance mode (type: Input).

0	Composition is constant	Composition is constant through the time window.
1	None	The composition of the inlet and outlet streams are not related.
2	Composition is variable (outlet = final)	Composition is variable over the time window. The composition of the outlet stream is assumed to be equal to the composition of the tank at the end of the time window.
3	Composition is variable (outlet is free)	Composition is variable over the time window. The composition of the outlet stream is not assumed to be equal to the final composition at the end of the time window. The user has to specify the final composition or the outlet composition to calculate the balance.

The **Composition** page allows keeping track of the initial and final composition of the tank. This page is only used when COMPMODE = 2:

Name: TK-1 / PFD1Category: StorageType: TNKVAL

GeneralParametersTags...ReactionsControl TagsOthersFlex codeCompositionStrapping

Compounds	Mass Initial	Mass%	Mass Final	Mass%	Delta
	kg		kg		kg
H2O	43544.100000	30.9561	245050.000000	30.1236	201505.900000
HNO3	97120.100000	69.0440	568433.000000	69.8764	471312.900000

Total

140664.000000

kg

813483.000000

kg

672819.000000

kg

OK

Cancel

Help

TABLE 41 - THE COMPOSITION PAGE OF A TNKVAL UNIT

NTABLE (General)

Strapping Table (type: Input).

0	None	No strapping table
1	Used	A strapping table is used

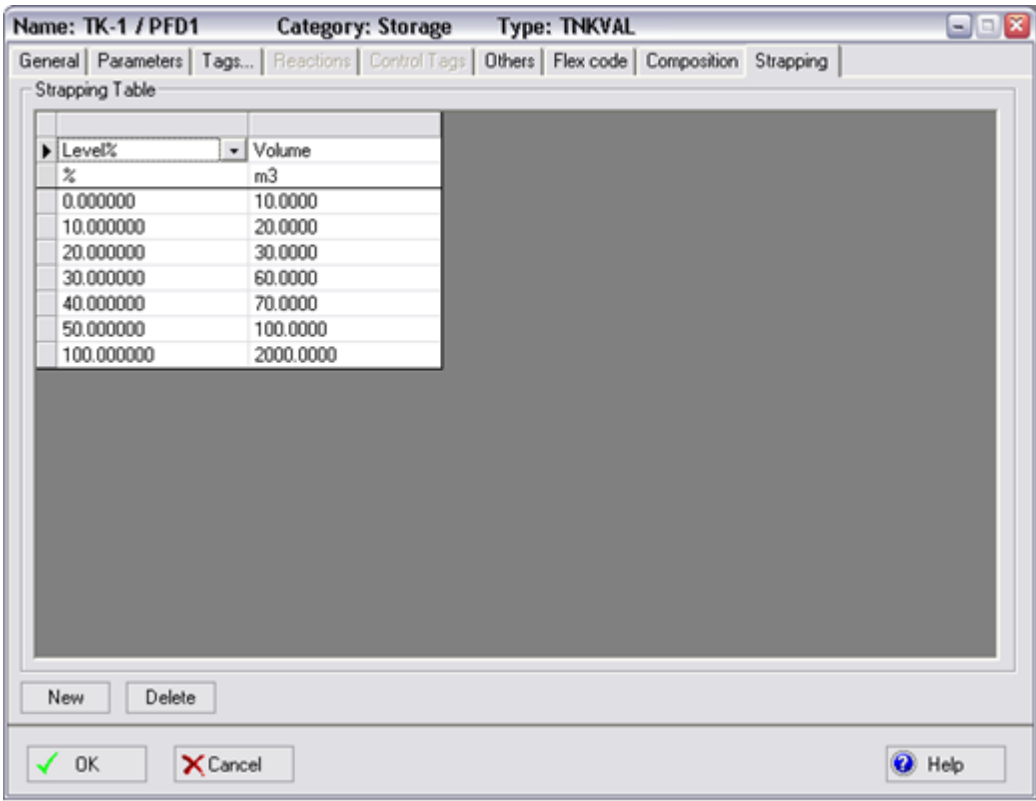


TABLE 42 - THE STRAPPING PAGE OF A TNKVAL UNIT

MODECURV (General)

Interpolation method (type: Input).

0	Linear interpolation	Linear Interpolation between the points of the strapping table
1	Spline function	Interpolation through a spline function between the points of the strapping table

OPTMODE (General)

Validation method (type: Input).

0	Level used as variable	The tank level will be defined as a variable and linked to the tank volume or mass by interpolation into the strapping table all along the calculation (reconciliation) phase
1	Volume/Mass used as variable	The measured tank level will be converted into a measured volume or mass before the calculation (reconciliation) phase. This measured volume or mass will be used for the calculation and converted back into a reconciled level after the reconciliation phase. This approach tends to be more stable as it avoids possible discontinuity in the derivatives of the tank volume or mass versus the level because of the interpolation process. You need to create the corresponding volume of mass tag and set it OFF before using this mode

INITDENS (General)

Density initialization mode (type: Input).

0	Based on tag data	
1	Based on product code	When based on product codes, the LL and HH filter bounds on density tags will set on product default values (set by the PRODUCT syntax in measurement files)

IMBMODE (General)

Imbalance mode (type: Input).

0	No Imbalance	No imbalance is considered
1	Imbalance computed	This mode enables an imbalance on the storage tank, so that a systematic bias between the validated and measured value can be suppressed by acting a loss or surplus of the tank

DMASS (Parameters)

Accumulation (Delta Mass) (type: Measurable).

$DMASS = MASS2 - MASS1$

DTIME (Parameters)

Time step (type: Input).

$DTIME = t2 - t1$

This parameter must always be specified.

WIMBAL (Parameters)

Weight Imbalance (type: Measurable).

VIMBAL (Parameters)

Volume Imbalance (type: Measurable).

These parameters are the computed imbalances on the tank. They will always be zero when the parameter IMBMODE is set to zero (No Imbalance).

MASS1,2 (Others)

Mass in tank at time t1 and t2 (type: Measurable).

VOL1,2 (Others)

Volume in tank at time t1 and t2 (type: Measurable).

DENS1,2 (Others)

Density in tank at time t1 and t2 (type: Measurable).

LEVEL1,2 (Others)

Level at time t1 and t2 (type: Measurable).

LVLFRA1,2 (Others)

Level Fraction at time t1 and t2 (type: Measurable).

When the volume mode is activated, all inventory variables (masses, volumes and densities) can be measured.

The level variables are only used when a strapping table is defined (NTABLE = 1). Depending on the way you define the strapping table, the level will be measured as a height (LEVEL) or as a percentage (LVLFRA).

SECTION 45: TURVAL : TURBINE

PURPOSE

This type of UNIT allows to consider the energy balance of a turbine.

Information on the isentropic (or polytropic) efficiency and the mechanical power can also be considered as measurements. Moreover a Stodola coefficient can be defined (or measured) to link the flow rate through the turbine to the driving pressure drop.

SYMBOLS

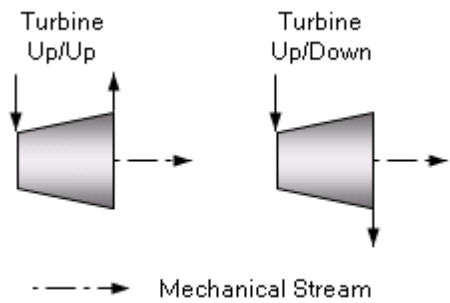


FIGURE 66 - SYMBOLS FOR UNIT TURVAL

CONNECTION RULES

INLET STREAMS

- 1 material

OUTLET STREAMS

- 1 material
- 1 mechanical

THERMOD

Both material streams must use the same THERMOD.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account

NAME	TAB	TYPE	DESCRIPTION
MODESTOD	General	Input	Selection of Stodola equation
MODEDP	General	Input	Selection of pressure equation
EFFIC	Parameters	Measurable	Isentropic efficiency
POLYEFFI	Parameters	Measurable	Polytropic efficiency
STODOLA	Parameters	Measurable	Stodola coefficient
DP	Parameters	Measurable	Pressure drop
PRATIO	Parameters	Measurable	Pressure ratio

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	All equations	Mass and energy balances are taken into account. The isentropic efficiency equation is generated as well
1	Mass Balance only	Only mass balances are taken into account
2	Mass and Energy Balance	Mass and energy balances are taken into account

MODESTOD (General)

Selection of Stodola equation (type: Input).

0	(none)	No Stodola equation
1	Classical Equation	$DP = \text{Stodola} \frac{G_{in}^{EXP} p_{in} v_{in}}{(p_{in} + p_{out})}$ where: ■ G_{in}: is the stream mass flow rate (kg/s) ■ p_{in} , p_{out}: are the inlet and outlet stream pressure (bar) ■ v_{in}: is the inlet stream volumetric weight (m³/kg) ■ Stodola: is the Stodola coefficient ■ EXP=2:
2	CMI's equation	Simplified equation (valid when outlet pressure is negligible): $\text{Stodola} = \frac{G_{in} \sqrt{R \cdot T_{in}}}{p_{in}}$

The Stodola equation links the pressure drop around the turbine to the flow rate.

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

EFFIC (Parameters)

Isentropic efficiency (type: Measurable).

This parameter is only relevant in MODE 0.

POLYEFFI (Parameters)

Polytropic efficiency (type: Measurable).

When negative, this parameter is ignored. When positive, an additional equation is generated to calculate the polytropic efficiency.

Attention: the formulation of the polytropic efficiency assumes that the mixture behaves like an ideal gas.

STODOLA (Parameters)

Stodola coefficient (type: measurable).

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{in} - P_{out}$$

This equation is not generated if MODEDP = 0.

where:

- P_{in} : the pressure of the inlet stream
- P_{out} : the pressure of the outlet stream

PRATIO (Parameters)

Pressure ratio (type: Measurable).

EXAMPLE : STEAM TURBINE

Attention: see model turval.bls in the directory "examples\turval". Units of type TURVAL are also used in the example coal.bls (coal fired power plant).

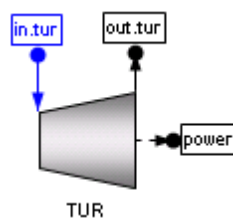


FIGURE 67 - TURVAL EXAMPLE

There is 1 UNIT:

- TUR
 - type=TURVAL
 - MODE='All equations'

3 STREAMS must be defined:

STREAMS THERMOD

CONNECTIONS

IN.TUR	STEAM	to TUR
OUT.TUR	STEAM	from TUR
POWER	MECHANICAL STREAM	from TUR

The THERMOD STEAM is defined in the file Empty.bls.

The list of measurements and results are given in tables below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
IN.TUR_MASSF	STREAM	IN.TUR	MASSF	50.000	-5.000	t/h
IN.TUR_P	STREAM	IN.TUR	P	12.000	0.100	bar
IN.TUR_T	STREAM	IN.TUR	T	240.00	1.00	C
OUT.TUR_P	STREAM	OUT.TUR	P	4.0000	0.100	bar
OUT.TUR_T	STREAM	OUT.TUR	T	147.00	1.00	C
POWER_POWER	STREAM	POWER	POWER	2180.0	-0.5000	kW
TUR EFFIC	UNIT	TUR	EFFIC	75.000	10.0	%

TABLE 43 - MEASUREMENTS FOR TURVAL EXAMPLE

Number of tags	7					
Number of equations	9					
Number of unmeasured variables	7					
Number of measured variables	7					
Number of variables considered constant	0					
Obvious number of redundancies	2					
Total number of redundancies	2					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	1.24697					
Chi-square test value	5.99000					
Sum of squares of residues	0.573703E-008					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
IN.TUR_MASSF	50.000	5.00 %	47.411	1.93 %	1.07	t/h
IN.TUR_P	12.000	0.100	12.005	0.997E-01	0.00	bar
IN.TUR_T	240.00	1.00	239.72	0.966	0.08	C
OUT.TUR_P	4.0000	0.100	3.9891	0.979E-01	0.01	bar
OUT.TUR_T	147.00	1.00	147.27	0.967	0.07	C
POWER_POWER	2180.0	0.500 %	2181.1	0.498 %	0.01	kW
TUR_EFFIC	75.000	10.0	75.180	2.31	0.00	%

TABLE 44 - RESULTS OF TURVAL EXAMPLE

SECTION 46: VARVAL : USER DEFINED VARIABLES

PURPOSE

A unit of type **VARVAL** gives access to 9 user defined variables which can be used in a Flex code or in an OPXVAL unit.

The values of these variables are available in the parameters VAR1...9 of the unit and can be measured. These parameters are dimensionless.

SYMBOLS



FIGURE 68 - SYMBOLS FOR UNIT VARVAL

CONNECTION RULES

No connections are allowed on a **VARVAL**.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
VARi	Parameters	Measurable	Additional variables (i=1..9)

SECTION 47: WALVAL : HEAT LOSSES THROUGH UNIT WALLS

PURPOSE

This type of UNIT is used to account for heat losses through external walls of a vessel. The losses are estimated on basis of the external or internal wall temperature. In the latter case, the wall thermal resistance must be provided. In both cases, the wall temperatures will be validated.

SYMBOLS

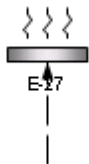


FIGURE 69 - SYMBOLS FOR UNIT WALVAL

CONNECTION RULES

INLET STREAMS

- 1 thermal stream

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
TAMB	Parameters	Input	Ambient temperature
AVERT	Parameters	Input	Vertical surface area
RVERT	Parameters	Input	Thermal resistance of Vertical wall
AHU	Parameters	Input	Horizontal surface area, facing up
RHU	Parameters	Input	Thermal resistance of Horizontal wall, facing up
AHD	Parameters	Input	Horizontal surface area, facing down
RHD	Parameters	Input	Thermal resistance of Horizontal wall, facing down
TWVEX	Parameters	Measurable	Vertical Wall external temperature

NAME	TAB	TYPE	DESCRIPTION
TWHUEX	Parameters	Measurable	Horizontal wall (facing up) external temperature
TWHDEX	Parameters	Measurable	Horizontal wall (facing down) external temperature
TWV	Parameters	Measurable	Vertical Wall temperature
TWHD	Parameters	Measurable	Horizontal wall (facing down) temperature
TWHU	Parameters	Measurable	Horizontal wall (facing up) temperature

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input).

0	Energy Balance	The energy balance is calculated
1	No Equation	No equation is generated

TAMB (Parameters)

Ambient temperature (type: Input).

This is the ambient temperature to be used for the heat loss calculation.

This value will not be validated during the run : it is a constant.

AVERT (Parameters)

Vertical surface area (type: Input).

AHU (Parameters)

Horizontal surface area, facing up (type: Input).

AHD (Parameters)

Horizontal surface area, facing down (type: Input).

These parameters respectively represent the vertical, horizontal facing up and horizontal facing down areas considered in order to calculate the heat loss.

RVERT (Parameters)

Thermal resistance of Vertical wall (type: Input).

RHU (Parameters)

Thermal resistance of Horizontal wall, facing up (type: Input).

RHD (Parameters)

Thermal resistance of Horizontal wall, facing down (type: Input).

These parameters represent the thermal resistance associated to each area and only have to be provided if the heat loss is calculated from the internal wall temperature.

TWVEX (Parameters)

Vertical Wall external temperature (type: Measurable).

TWHUEX (Parameters)

Horizontal wall (facing up) external temperature (type: Measurable).

TWHDEX (Parameters)

Horizontal wall (facing down) external temperature (type: Measurable).

These parameters are displayed for information and represent the external wall temperature.

They are useful when the thermal wall resistance is considered.

TWV (Parameters)

Vertical Wall temperature (type: Measurable).

TWHD (Parameters)

Horizontal wall (facing down) temperature (type: Measurable).

TWHU (Parameters)

Horizontal wall (facing up) temperature (type: Measurable).

These parameters are either the external or the internal wall temperatures. In the latter case parameters TW.xxEX and Rxx are used.

EXAMPLE: HEAT LOSSES IN A VERTICAL PIPE

Attention: see model walval.bls in the directory "examples\walval".

This example deals with the heat loss of a vertical pipe. This heat loss is estimated from the external

wall temperature, so that the thermal resistance does not have to be provided.

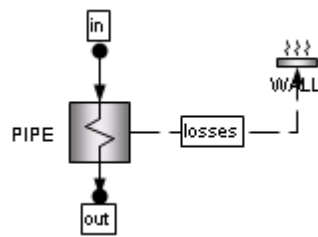


FIGURE 70 - AN EXAMPLE OF WALVAL TYPE UNIT

There are 2 UNITS:

- PIPE
 - type=EEXVAL
 - MODE='Mass and Energy Balance'
- WALL
 - type=WALVAL
 - AVERT=40 m2 (Vertical plate area)
 - TAMB=25 C (Ambient Temperature)

3 STREAMS must be defined:

STREAMS	THERMOD	CONNECTIONS
IN	STEAM	to PIPE
OUT	STEAM	from PIPE
LOSSES	THERMAL STREAM	from PIPE to WALL

The THERMOD STEAM is available in the model file Empty.bls.

The list of measurements and results are given in tables below:

TAGS DEFINITION						
Name	Object	Owner	Var. Name	Mea. val.	Mea. acc.	Unit
IN_MASSF	STREAM	IN	MASSF	100.00	-5.000	kg/s
IN_P	STREAM	IN	P	30.000	0.100	bar
IN_T	STREAM	IN	T	350.00	2.00	C
LOSSES_LOAD	STREAM	LOSSES	LOAD	300.00	-10.00	kW
OUT_P	STREAM	OUT	P	29.000	0.100	bar
OUT_T	STREAM	OUT	T	348.00	2.00	C
PIPE_DP1	UNIT	PIPE	DP1	0.80000	0.100	bar
WALL_TWV	UNIT	WALL	TWV	300.00	5.00	C

TABLE 45 - MEASUREMENT LIST FOR WALVAL EXAMPLE

Number of tags	8					
Number of equations	6					
Number of unmeasured variables	3					
Number of measured variables	8					
Number of variables considered constant	0					
Obvious number of redundancies	3					
Total number of redundancies	3					
Number of trivial redundancies	0					
Validation and reconciliation results						
Objective function	1.43223					
Chi-square test value	7.81000					
Sum of squares of residues	0.176804E-014					
Number of active bounds	0					
Number of variables close to bounds	0					
Number of eliminated tags	0					
Number of flagged tags	0					
Number of filtered tags	0					
TAGS IN THE MODEL						
Tag name	Mea. Val.	Mea. acc.	Rec. Val.	Rec. acc.	Penalty	Unit
IN_MASSF	100.00	5.00 %	100.00	5.00 %	0.00	kg/s
IN_P	30.000	0.100	29.933	0.816E-01	0.45	bar
IN_T	350.00	2.00	350.06	1.41	0.00	C
LOSSES_LOAD	300.00	10.0 %	308.82	3.18 %	0.09	kW
OUT_P	29.000	0.100	29.067	0.816E-01	0.45	bar
OUT_T	348.00	2.00	347.94	1.42	0.00	C
PIPE_DP1	0.80000	0.100	0.86657	0.816E-01	0.44	bar
WALL_TWV	300.00	5.00	299.49	4.72	0.01	C

TABLE 46 - RESULTS FOR WALVAL EXAMPLE

SECTION 48: WLEVAL : WATER-LIQUID ORGANIC SEPARATOR

PURPOSE

The **WLEVAL** unit models water decanting from an organic solution.

SYMBOLS

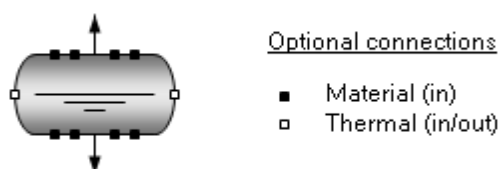


FIGURE 71 - SYMBOLS FOR UNIT WLEVAL

CONNECTION RULES

INLET STREAMS

- Maximum number:
 - Material: 8
 - Thermal: 2

OUTLET STREAMS

- Maximum number:
 - Material: 2 streams of state L, LL or WL
 - Thermal: 2

THERMODS

Both outlet streams must use the same THERMOD.

Attention:

1. The state of the outlet material streams can also be LV, LLV and WLV provided that there is more than one component in the stream.
2. Water is assumed to be pure and to be the heavy liquid stream and is therefore going through the lower outlet stream. The solubility of water in the organic phase is based on the solubility in kerosene.
3. Solids that would be present in the inlet streams will always end up in the organic phase.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
FLASH	General	Input	Flash initialization
REFDP	Parameters	Input	Reference stream for pressure drop
DP	Parameters	Measurable	Pressure drop

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	All equations	Mass, energy balances and liquid-liquid equilibrium relationships are taken into account
1	Mass Balance only	Only the mass balances are taken into account
2	Mass and Energy Balance	Mass and energy balances are taken into account
3	Mass and W/L equilibrium	Mass and water-liquid equilibrium relationships are taken into account

Attention:

1. The outlet streams are always set at the same pressure and temperature except in Mode 1 (Mass balance only).
2. The water solubility in the organic phase is a function of temperature:

$$\text{EXP} \left(X_{\text{WAT0}} - \frac{X_{\text{WAT1}}}{T} \right)$$

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

FLASH (General)

Flash initialization (type: Input).

0	None	no initialization of outlet streams (keep values of last run)
1	Always	initialization in all cases on basis of input streams data
2	When needed	initialization on basis of input streams data in the following cases: ▪ one of the flow rates of the two first outlet streams is equal to 0 ▪ the temperature of one of the two first outlet streams is equal to 0°C

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the referece inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if $MODEDP = 0$.

SECTION 49: WLVAL : LIQUID-LIQUID-VAPOR SEPARATOR

PURPOSE

This type of UNIT handles mass balances, energy balance and water-liquid-vapor equilibrium relationships of a 3 phase separator with decantation of water as one of the liquid phases.

SYMBOLS

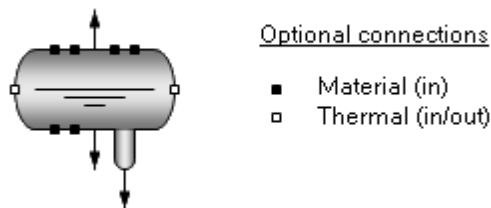


FIGURE 72 - SYMBOLS FOR UNIT WLVAL

CONNECTION RULES

INLET STREAMS

- Maximum number:
 - Material: 6
 - Thermal: 2

OUTLET STREAMS

- Maximum number:
 - Material:
 - 2 streams of state L, LL or WL for the liquids
 - 1 stream of state V
 - Thermal: 2

THERMODS

All outlet streams must use the same THERMOD.

Attention:

1. The state of the outlet material streams can also be LV, LLV and WLV provided that there is more than one component in the stream.

2. It is essential to initialize the composition of the outlet streams before attempting to activate the L/L/V equations. Failure to do so can induce convergence problems or lead to a trivial solution where all outlet streams have the same compositions. The initialization can be performed manually or automatically via the parameter FLASH.
3. Water is assumed to be pure and to be the heavy liquid stream and is therefore going through the lower outlet stream. The solubility of water in the organic phase is based on the solubility in kerosene.
4. No assumption is made about the destination of solids that would be present in the inlet streams. Composition measurements (or information coming from the downstream balances) should define whether solids are going with the liquid (organic) or vapor phase.

UNIT PARAMETERS (SUMMARY)

NAME	TAB	TYPE	DESCRIPTION
MODE	General	Input	Selection of equations to be taken into account
MODEDP	General	Input	Pressure equation
FLASH	General	Input	Flash initialization
REFDP	Parameters	Input	Reference stream for pressure drop
DP	Parameters	Measurable	Pressure drop

UNIT PARAMETERS

MODE (General)

Selection of equations to be taken into account (type: Input)

0	All equations	Mass, energy balances and liquid-liquid equilibrium relationships are taken into account
1	Mass Balance only	Only the mass balances are taken into account
2	Mass and Energy Balance	Mass and energy balances are taken into account
3	Mass and W/L/V equilibrium	Mass and water-liquid-vapor equilibrium relationships are taken into account

Attention:

1. The outlet streams are always set at the same pressure and temperature except in Mode 1 (Mass balance only).

2. If the partial flow rate of some of the compounds is detected as being equal to zero in one of the Streams, the related equilibrium relationship is not considered.

MODEDP (General)

Pressure equation (type: Input).

0	None	No pressure drop equation will be generated
1	DP variable	A pressure drop equation will be generated

FLASH (General)

Flash initialization (type: Input).

0	None	no initialization of outlet streams (keep values of last run)
1	Always	initialization in all cases on basis of input streams data
2	When needed	initialization on basis of input streams data in the following cases: ▪ one of the flow rates of the two first outlet streams is equal to 0 ▪ the temperature of one of the two first outlet streams is equal to 0°C

REFDP (Parameters)

Reference stream for pressure drop (type: Input).

0	All inlets	All inlet streams are set at the same pressure
1, 2, 3, etc.	<stream>	Indicates which inlet stream is used as reference for the pressure drop equation. The pressures of the inlet streams are not related

DP (Parameters)

Pressure drop (type: Measurable).

Calculates pressure drop through the unit, according to following equation:

$$DP = P_{ref} - P_{out}$$

where:

- P_{ref} : the pressure of the reference inlet stream
- P_{out} : the pressure of the outlet stream

This equation is not generated if $MODEDP = 0$.



belsim

CHAPTER 7

FLEX LANGUAGE DESCRIPTION

SECTION 1: INTRODUCTION

1.1 WHAT YOU CAN DO WITH FLEX

Flex is a programming language, which allows users to define and obtain the computation of mathematical expressions. In a VALI application, Flex programs can be used:

- in the filtering phase:
 - to define advanced filtering operations (ValiFilter license required)
- in the calculation phase:
 - to add user-defined constraints
 - to add user-defined measurements (Unit FLXVAL, VARVAL or FLEXLIB)
 - to calculate user defined functions of the validated values
 - to specify user defined kinetics (Unit RKIVAL)
- in the report phase:
 - to write specific reports

Using Flex language, you can access values of streams and units and perform various calculations. Thermodynamic routines can be used to calculate various thermodynamic properties and perform physical equilibrium compositions. You can define your own subroutines and refer them from different units.

For VALI application, Flex programs can be associated to:

- any units representing an equipment, for adding filtering operations, constraints and reports
- units of type FLXVAL for adding filtering operations, calculating values of the parameters of the unit (var1, var2, etc) which can then be measured by creating a tag on them, and generating reports

The section about [the UNITS](#) indicates how to access the Flex program.

The section dealing with "[FLXVAL](#)" describes how to use such units and gives a practical example.

This chapter describes the general syntax of the language Flex as well as the specific functions available in the VALI implementation of Flex.

1.2 INTERPRETED OR COMPILED

A Flex program is either:

- interpreted at runtime, each time the program is executed during a run of a model
- interpreted at runtime during the verification phase and executed each time the program is executed during a run

The latter mode is quicker and is used by default. The first mode is only kept for compatibility reason and can be reactivated using the parameter `FLEXMODE` of unit `OPTVAL`.

SECTION 2: FLEX LANGUAGE OBJECTS

FLEX programs are built by using character strings defining words, instructions and lines.

The length of a Flex line is limited to 510 characters, while words and instructions are limited to 32 characters.

A line is made with a variable number of characters. It may contain one or more instructions, separated by a ";". Instructions may not be split over different lines.

The exclamation mark "!" means the beginning of a comment, then carrying on up to the end of the line. A "!" in a line forces (means) the end of the preceding instruction.

The character set used by FLEX is defined by :

- letters : capital and lower-case, which are differentiated only in printed text.
 - UPPER-CASE :A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
 - lower-case :a b c d e f g h i j k l m n o p q r s t u v w x y z
 - numbers :0 1 2 3 4 5 6 7 8 9
 - special characters :+ - * / ^ = > < # & | ? ' () , . \ ; ! \$
- space (ASCII 32 characters)
- separator function (not ignored as in FORTRAN)

SECTION 3: LANGUAGE ELEMENTS

Instructions are made up with 5 types of elements:

- separators
- constants
- variables
- operators
- keywords

Operators are represented by particular characters or reserved keywords.

3.1 SEPARATORS

Separators are particular characters or spaces.

The **BLANK** (spaces). You may have more than one space as a separator between two elements of code. Blank is significant in character string constants (i.e. in string 'Results table').

The **SEMICOLON** is used to separate instructions written on the same line. An instruction may not be split between lines.

The **COMMA** is used to separate elements of a function arguments list, or the elements of a printing list (WRITE instruction).

The **BRACKETS** surround a function arguments list or a table index. The brackets also permit changes in the order of calculation of complex relations.

The **EXCLAMATION MARK** means that the end of the line is a comment.

The **EQUALS SIGN** (=) or **COLON** (:) is used for the affectation instruction. It separates the name of the target variable and the result to be stored.

```
X=3 ;           X:3;
```

Both instructions store 3 in X .

3.2 CONSTANTS

Constants can be numerical values, elements for reference to an object (NAME), or character strings.

FLEX recognizes and represents numerical constants, as in the programming language FORTRAN:

- optional sign, integer, decimal point, decimal, "E" symbol to define exponent to 10 and integer exponent which may be preceded by a sign and have a range from -35 and up to +35
- integer, or decimal part is optional. When the decimal part is omitted, the decimal point could then be omitted. Exponent is optional

Thus, the following numbers are recognized numerical constants:

```
1      +1      1      2.      .2      +2.      .2
1.2    2.03    2.E+03 2.5E 4 +2.E3    .2E 5 2E4
```

FLEX is written in FORTRAN and all of the computations are with real numbers (at least 12 significant digits).

Numerical constants are stored in variables of such a precision.

Character strings may only be in the FLEX program if they are constant. They consist in an arbitrary number of characters between two apostrophes. Therefore, the apostrophe is the only character, which cannot be in a string of characters.

Constants of the NAME type are strings with 1 to 32 characters, without any spaces.

```
Integer_ = number [number]
```

```
not_sign
ed
```

```
Integer_c= [+ | -] Integer_without_sign
onstant
```

```
Real_not = Integer_not_signed.[Integer_not_signed] | . Integer_not_signed
_signed
```

```
Real_bas = [+ | -] Real_not_signed
ic
```

```
Real_con = Real_basic [E Integer_constant]
stant
```

```
Characte = ' Character_Element [Character_Element ] '
r_consta
nt
```

3.3 VARIABLES

A variable is identified by its name, which is a string of 32 characters maximum (letters or numbers). The first character must be alphabetic, but the code is not case-sensitive: capitals and lower-cases are not distinguished in names.

Variables and operators may not share the same names.

Valid names for variables are: A12B, GHIJKLMN, PI314

There are three different types of variables:

- real (REAL)
- integer (INTEGER)
- reference to an object by its name (NAME)

Variables may be scalar or of an array type (table with one index).

The type of variable and its potential index are notified to FLEX by non-executable declaration instructions.

These instructions are made up with a keyword defining the type of variable (INTEGER, REAL or NAME), followed by a list of variable names to be assigned.

Every variable not explicitly declared is supposed to be a real scalar variable and is created when first used.

The initial values of numerical variables are zero and as far as character strings and variables of type NAME are concerned, it is ''.

Logical states are stored in temporary numerical variables:

- TRUE state corresponds to 1
- FALSE state corresponds to 0

Program variables can be accessed by all called functions.

Variables created (explicitly or implicitly) in a function are eliminated after the execution of that function.

Existing variables cannot be declared again.

To declare real variables, following syntax can be used:

REAL XXX (N) , YYY

where:

- XXX (N) declares an array. XXX can be any name, N will be the size of array XXX
- YYY declares a scalar. YYY can be any name

Example:

```
REAL VAP(3), LIQ(3), T, P
```

To declare integer variables, following syntax can be used:

INTEGER XXX (N) , YYY

where:

- XXX (N) declares an array. XXX can be any name, N will be the size of array XXX
- YYY declares a scalar. YYY can be any name

Example:

```
INTEGER INDEX(3), LISTE(3), S1, K3(3*2-4)
```

To declare string variables, following syntax can be used:

NAME XXX (N) , YYY

where:

- XXX (N) declares an array. XXX can be any name, N will be the size of array XXX
- YYY declares a scalar. YYY can be any name

Example:

```
NAME NAMES (3) , ANAME
```

3.4 OPERATORS AND INTERNAL FUNCTIONS

Operators define basic operations that can be made with variables and constants. Expressions are made up of one or more operator applied on variables, constants or expressions:

- logical operators:
 - `?` or `.NOT.`: logical negation
 - `&` or `.AND.`: logical AND
 - `|` or `.OR.`: logical OR
- numerical comparison operators:
 - `=` or `.EQ.`: equal
 - `#` or `.NEQ.`: not equal
 - `>` or `.GT.`: greater than
 - `<` or `.LT.`: lesser than
 - `>=` or `.GE.`: greater or equal
 - `<=` or `.LE.`: lesser or equal
- numerical operators:
 - `+`: addition
 - `-`: subtraction
 - `*`: multiplication
 - `/`: division
 - `^` or `**`: power
- numerical functions:
 - **FRAC** (X) : fractional part
 - **INT** (X) : entire part
 - **ABS** (X) : absolute value
 - **SQRT** (X) : square root
 - **EXP** (X) : exponential
 - **LOG** (X) : natural logarithm
 - **LOG10** (X) : logarithm base 10
 - **SIN** (X) : sine (argument in radian)
 - **COS** (X) : cosine (argument in radian)
 - **TAN** (X) : tangent (argument in radian)
 - **TANH** (X) : hyperbolic tangent

- **ASIN** (X) : inverse sine
- **ACOS** (X) : inverse cosine
- **ATAN** (X) : inverse tangent
- **LEN** (X) : dimension of array variable

3.5 OPERATIONS EXECUTION ORDER

The functions are evaluated in the following order:

- functions calls
- power (^)
- multiplication and division (* /)
- addition and subtraction (+ -)
- comparisons (> < = # <= >=)
- logical negation (?)
- logical AND (&)
- logical OR (|)

Operations of the same priority are evaluated from left to right.

The use of brackets allows users to modify the evaluation order, as the contents of inner brackets are evaluated before the others. In any expression, the number of open brackets must be equal to the number of closed ones.

Examples:

```
1
A+3
(-B+sqrt(B**2-4*A*C))/2/A
a=b&c>D.or.x(5)>=b^2-4*A*C
```

Terms of logical operators could be numerical expressions.

Expressions equal to zero are interpreted as FALSE logical expressions, else, TRUE.

Results of a logical expression could be used as numerical variables:

- TRUE conditions are represented by a value of 1
- FALSE conditions are represented by a value of 0

3.6 RESERVED KEYWORDS

FLEX uses some reserved keywords that cannot be used to name variables. These keywords are used to build executable instructions and declarations.

Reserved keywords are:

- type declaration (non-executable):
 - **REAL**
 - **INTEGER**
 - **NAME**
 - **FUNCTION**
- input-output instructions (executable):
 - **READ**
 - **WRITE**
 - **TAB**
 - **EDIT**
- control of the computation sequence (executable):
 - **CALL**
 - **RETURN**
 - **END**
 - **WHILE**
 - **ENDWHILE**
 - **UNTIL**
 - **ENDUNTIL**
 - **IF**
 - **THEN**
 - **ELSEIF**
 - **ELSE**
 - **ENDIF**
- internal functions:
 - **OPENIN**
 - **OPENOUT**
 - **DEBUG**
 - **TSTFUN**
- internal functions defined by the application associated with FLEX

SECTION 4: EXECUTABLE INSTRUCTIONS

Executable instructions are simple instructions or blocks of instructions.

FLEX executes instructions sequentially, but it is possible to pack the instruction into blocks with a control instruction. These instructions also allow you to control the sequence by using logical tests.

Instruction blocks are surrounded by couples of associated keywords:

```
WHILE ... ENDWHILE
UNTIL ... ENDUNTIL
IF ... THEN ... ELSEIF ... ELSE ... ENDIF
```

4.1 AFFECTATION

Stores the result of expression computation in a variable. FLEX is able to compute complex expressions, syntax and priority in computation, which has already been discussed.

Examples:

```
A=2
B (2*j+5) = 3* (A-6*C/B (3) +SQRT (D-4) )
```

4.2 READ

Reads the value of a variable in a file or with a request on screen; in this case, a message with the name of the variable to set is printed on screen. It is also possible to define a file to enter that value with the use of OPENIN function.

Examples:

```
READ A
READ B (2*j-5)
```

4.3 WRITE

Prints the value of a variable or expression in a file or on screen. The destination of a write instruction could be changed with the OPENOUT function. All the values of the same WRITE instruction are printed on the same line if possible. The defaults for FLEX are the elimination of redundant zeros of numerical values and the introduction of a space after each value. Each WRITE instruction is followed by a jump to the next line, except if the list is ended by a colon (;).

A WRITE instruction without any arguments leads to a blank line.

TAB and EDIT keywords are used to modify the presentation of results.

TAB(Expression) prints the next variable in the column corresponding to the value of that expression.

EDIT(format), followed by an expression to be printed, prints the value of the expression with the FORTRAN FORMAT, without elimination of the redundant zeros by FLEX.

EXAMPLES	RESULT ON SCREEN
A=2.5	
WRITE A	2.5
WRITE 2*A	5
WRITE A :	
WRITE 2*A	2.5 5
WRITE 'Result :',3*A,'/',10.00	Result : 7.5 / 10
WRITE A,TAB(4*A),1/A	2.5 0.4
WRITE EDIT(E12.5)A	0.25000E+01
WRITE tab(3),EDIT(E12.5)A	0.25000E+01
WRITE tab(3),EDIT(2X,E12.5)A	0.25000E+01

4.4 WHILE BLOCK

Following syntax is used for **WHILE** block:

WHILE LOGICAL EXPRESSION

{ INSTRUCTIONS }

ENDWHILE

If the logical expression test is TRUE, the instructions contained in the WHILE and ENDWHILE keywords are executed and restarted "while" the condition is TRUE.

If the logical expression test is FALSE, go to the next instruction following ENDWHILE.

Blocks might never be used.

4.5 UNTIL BLOCK

Following syntax is used for **UNTIL** block:

UNTIL LOGICAL EXPRESSION

{ INSTRUCTIONS }

ENDUNTIL

Flex EXECUTES the instructions ONCE until the ENDUNTIL keyword; then it estimates the logical

expression.

If the result is TRUE, it goes to the instruction which follows the ENDUNTIL keyword.

If the result is FALSE, it returns to the first instruction of the UNTIL block.

Thus, the instruction block is executed once at least.

4.6 IF BLOCK

Following syntax is used for **IF** block:

```
IF LOGICAL EXPRESSION THEN
    { INSTRUCTIONS }
[ ELSEIF LOGICAL EXPRESSION THEN
    { INSTRUCTIONS } ]
[ ELSE
    { INSTRUCTIONS } ]
ENDIF
```

ELSEIF and **ELSE** are optional commands. The **ELSEIF** instruction can be repeated.

FLEX estimates the logical expression following **IF**:

- if it is TRUE, then it executes the instructions preceding the next **ELSEIF**, **ELSE** or **ENDIF** and then proceeds to the next instruction following the corresponding **ENDIF** keyword
- if the logical expression is FALSE, it looks for an **ELSEIF** statement with a true logical expression to proceed in the same way
- if no **IF** or **ELSEIF** statement with a true logical function is encountered, FLEX then looks for the **ELSE** instruction and executes the instructions until the **ENDIF** keyword is found
- if no **ELSE** instruction is found, FLEX continues until the next **ENDIF** instruction is reached

Please note that instructions following keywords **THEN** or **ELSE** have to be separated by a semicolon or to be written on the next line, i.e.:

```
IF A>0 THEN;RES='positive';ELSEIF A=0 THEN
RES='null';ELSE;RES='negative';ENDIF
```

4.7 END

Ends the execution of a function and returns to the next instruction of the calling function.

END also terminates the function definition.

4.8 RETURN

Ends the execution of a function and returns to the next instruction of the calling function.

4.9 SYNTAX OF EXECUTABLE INSTRUCTIONS

Executable = Simple_instr | Block_inst
e_instr

Simple_inst = Affectation | Read_inst | Write_inst | Call_inst | Return_inst | End_inst
str

Affectation = Elem_variable = Expression
n

Elem_var = Variable_name | Variable_name(Index)
iable

Variable_ = Letter [Letter | Number]
name

Index = Arith_exp

Arith_exp = [+ | -] Arith_term { <+ | -> Arith_term }
p

Arith_term = Arith_factor { < * | / > Arith_factor }
m

Arith_fac = Arith_prim { ^ Arith_prim }
tor

Arith_pri = Elem_variable | Ref_function | Arith_constant | Arith_exp | Logical_exp
m

Arith_const = Integer_constant | Real_constant
nstant

Read_inst = READ Elem_variable

Write_inst = WRITE Write_list
t

Write_list = Write_elem [, Write_elem]
t

Write_elem = Arith_exp | TAB(Arith_exp) | EDIT(Format) Arith_exp
m

Format = any list of FORTRAN format acceptable to treat the "Arith_exp"

Return_inst = RETURN
nst

End_inst = END

Block_inst = Block_if | Block_while | Block_until
t

Block_if = Block_then { Block_elseif } [Block_else] ENDIF

Block_then = IF Logical_exp THEN { Exec_instr }
n

Block_els = ELSEIF Logical_exp THEN { Exec_instr }

SECTION 5: FLEX FUNCTIONS

5.1 USING FLEX FUNCTIONS

TYPES OF FLEX FUNCTIONS

There are three types of FLEX functions:

- internal FLEX functions
- application dependant functions: VALI, SIMU or other FLEX compatible programs give access to a number of specific functions. These functions are discussed in a separate section
- user defined functions: you can define your own functions and reuse them in different places

CALLING A FUNCTION

These three types of functions are called using the same syntax:

CALL A_FUNCTION (ARGUMENTS)

If the name of the function is recognised, FLEX computes the parameters (which might be constants, scalar variables, parts of arrays, or expressions), the number of these arguments is checked, and their type has to be in agreement with those used in the function called.

FLEX stops the program if an inconsistency is found.

After the execution of the function, the program continues with the next instruction within the calling module.

Examples:

```
CALL TSTFUN(A)
CALL OPENIN('MYFILE')
CALL REPORT(3,A(8), 3*B-4,EXP(1.5))
```

ARGUMENTS

Arguments are transferred to the function either by address or by value.

If the argument is a variable or an element of an array, then the address of the argument is transferred to the function. If the function modifies the argument contents, then these changes will be reflected in the variable of array element after the call of the routine.

If the argument is an expression, then FLEX creates temporary variables to store the results of the expressions. In such a case if the called function changes the value of an argument, the result will not be retrieved by the calling program.

In the following example, where TEST is a function affecting a value of 3 to its argument, it shows that:

```
CALL TEST(A)
WRITE A :
A=5
```

```
CALL TEST (-A)
WRITE A
END
FUNCTION TEST (X)
X=3
END
```

Printed result: 3 5

The variable A is modified during the first call. In the second call, whose argument is -A : an expression, thus FLEX creates a temporary variable to store the value -5 and use it as a parameter. This temporary variable is modified by the function, but A is not modified.

5.2 FLEX INTERNAL FUNCTIONS

5.2.1 DEBUG

This function is used to trace part of a program.

The following syntax can be used:

CALL DEBUG (I)

This is used along with a numerical argument.

If the argument is less than 1: no trace is printed.

If the argument is between 1 and 2, FLEX prints each instruction before execution.

If it is bigger than 2, a complete trace, with coding of the expressions in inverse polish notation, is printed.

Thus, it is possible to ask for a trace for part of the program.

Example:

```
CALL DEBUG (1)
```

5.2.2 TSTFUN

This function allows to test a function.

The following syntax can be used:

CALL TSTFUN (VAR)

This function is useful to check a program: it allows the user to print the name and the value of a scalar variable or of all the elements of an array.

Example:

```
CALL TSTFUN (A)
```

5.2.3 OPENIN

This function allows to open a file from which data can be read.

The following syntax can be used:

CALL OPENIN(FILE_NAME)

Uses a character string as its argument, in order to define the name of an existing file to be open for reading

In the following READ instructions, the values will be read in that file.

To reset the read target to the screen, call **OPENIN** with numerical argument equal to zero.

Example:

```
CALL OPENIN('MYFILE.DAT')
...
CALL OPENIN(0)
```

5.2.4 OPENOUT

This function allows to open a file to which data can be written.

The following syntax can be used:

CALL OPENOUT(FILE_NAME)

Uses a character string as its argument, in order to define the name of the file to print all WRITE instructions.

To reset the written target to the screen, call **OPENOUT** with a numerical argument equal to zero.

Example:

```
CALL OPENOUT('REPORT.LIS')
...
CALL OPENOUT(0)
```

5.2.5 CONCAT

This function allows to concatenate two strings.

The following syntax can be used:

CALL CONCAT(TEXT1, TEXT2, RESULT)

Joins the text strings text1 and text2 into one text string Result.

5.2.6 NAMEISEQUAL

This function allows to compare two strings.

The following syntax can be used:

CALL NAMEISEQUAL(TEXT1, TEXT2, RESULT)

TEXT1 and TEXT2 are the input strings you want to compare; RESULT is the output, it is a real. Its value will be 1 if the strings are equal and 0 otherwise.

RESULT can then be used as a condition in a block: a RESULT of 1 will be evaluated as "true", and a result of 0 will be evaluated as "false".

Notice that strings in flex are case-insensitive.

Example:

```
...  
CALL NAMEISEQUAL(TEXT1, TEXT1, RESULT)  
IF (RESULT) THEN  
WRITE 'these strings are equal'  
ELSE  
WRITE 'these strings are not equal'  
ENDIF
```

5.2.7 FLXDLL

This function allows the user to call user-defined subroutines in a FLEX code.

To use it, the user can replace the dynamic linked library flx.dll with his own library; the dll must contain the following subroutine:

SUBROUTINE (VARIN, VAROUT, PARAM, NUM, NIN, NOUT, NPAR)

It is then possible to call the subroutine in a FLEX code by using the command:

CALL FLXDLL (VARIN, VAROUT, PARAM, NUM, NIN, NOUT, NPAR)

where,

- VARIN: input, double precision array, dimension (NIN); The array of input variables for the user-defined subroutine
- VAROUT: output, double precision array, dimension (NOUT); The array of output variables for the user-defined subroutine
- PARAM: input, double precision array, dimension (NPAR); An array of parameters used by the user-defined subroutine
- NUM: An input integer; it can be used to allow the user to select the subroutine used between several (eg, with NUM=1, FLXDLL will compute the numerical derivative of the input array, and with NUM=2, it will compute the FFT of the input array)
- NIN: input, integer; Size of the array VARIN
- NOUT: input, integer; Size of the array VAROUT
- NPAR: input, integer; Size of the array PARAM

5.3 USER-DEFINED FUNCTIONS

In VALI, only functions can be defined (up to 9 functions generating equations per unit), each function defining the residual of one equation.

These functions can call other functions defined by the user. These user-defined functions can appear

in the Flex code page of any unit where a Flex code can be defined. They can be called from any other Flex code of the same model, when the compiled Flex option is activated, which is the case by default. If not, only functions defined in the same Flex code unit can be used. It is recommended to code user-defined functions in a Flexlib unit, so that the functions are available anywhere in the model.

Functions have the following lay-out:

```
FUNCTION NAME [ ( ARGUMENTS) ]
    FLEX INSTRUCTIONS...
END
```

where:

- NAME: the function name
- ARGUMENTS: the list of arguments, separated by a comma

When defining the functions calculating the residuals in VALI (FLXVAL units), the function name must respectively be EQ1, EQ2, etc. In that case, no argument is used

The type of arguments cannot be defined in the user-defined function but will instead take the type and dimension previously declared in the calling routine.

The function also inherits the entire environment defined in the calling routine. For example, routines used to get values from streams (GETVAR, GETALL, etc) can refer to the streams by their name or by their number. A positive (negative) number will correspond to an inlet (outlet) stream of the calling module.

This makes the writing of generic user-defined functions quite easy, ensuring maximum reusability of the code.

Return to the calling routine will be initiated by the RETURN command or when encountering the END of the function.

Recursive calls are allowed within the limits of available memory.

Example:

```
! Calling the function POLYNOM to calculate  $fx = 0.1 + 0.3 x - 0.5 x^2$ 
!
CALL POLYNOM (0.1, 0.3, -0.5, 0.1, fx)

FUNCTION POLYNOM (a, b, c, x, fx)
! User defined function to calculate a polynomial expression
Fx = a + b*x + c*x*x
RETURN
END
```

SECTION 6: EXAMPLE OF A FLEX PROGRAMME

This example is to show the principal control structures of FLEX, and does not claim to be a model of clarity or efficiency to estimate a factorial computation.

```
! FLEX test
! Factorial computation; reserved keywords are in capital
INTEGER n,j
REAL fac(50)
n=1
UNTIL n=0
    WRITE ' Enter a number, 0 to end' ; READ n
    IF n<0 THEN
        WRITE(' Positive please')
    ELSEIF n>0 THEN
        fac(1)=1.
        j=1
        WHILE j<n
            j=j+1
            IF j<= 50 THEN
                fac(j)=fac(j-1)*j
                WRITE ' The factorial of',J,' is',fac(j)
            ELSE ; WRITE ' Maximum 50'
                j=n
            ENDIF
        ENDWHILE
    ENDIF
ENDUNTIL
END
```

SECTION 7: VALI IMPLEMENTATION OF FLEX

7.1 INTRODUCTION

FILTERING

In the filtering phase, you can define specific actions to deactivate parts of a model or flex equations, or to change the measured value or precision of any measurement.

The corresponding Flex code can be defined on any unit and must follow the following syntax.

The specific functions available for the filtering are described in the paragraph **FLEX SUBROUTINES USED IN FILTERING** below.

FUNCTION FILTER

```
...your filtering code ...
END
```

FLEX EQUATIONS DURING CALCULATION PHASE

When using FLEX equations in calculation phase, each equation must be defined in a separate FUNCTION (EQ1, EQ2, ...) and the result of the calculations must be stored in the variable RES. The value of RES will automatically be available in the variables VARi of the unit (VAR1 for equation EQ1, etc.) on FLXVAL units or will be set to zero on other units.

The following internal variables are available:

- **PFDNAM**: name of the PFD where the unit is located (type NAME)
- **ITERFLEX**: number of iterations performed so far (type INTEGER)

Typically, each FUNCTION consists in:

- a declaration section: defines variables and arrays that will be necessary
- selection of a THERMOD: defines the thermodynamics methods to be used
- selection of variables: defines which variables will be used in the code
- calculations: defines the calculations and actions

Example:

```
FUNCTION EQ1 ! This the header for the first equation
! Declaration
REAL ...
! Selection of thermodynamic methods
CALL SELECT('Astream')
! Selection of variables
CALL GETALL(...)
CALL GETVAR(...)
CALL GETUVA(...)
CALL GTVARTAG(...)
CALL GTTAGVAL( )
! Calculations
...
```

```
RES = ...  
END
```

Attention: By default, if one of the units or streams referred by you flex code is OFF, then the equation is not generated at all. However, if you use the function GTSTATUS, the equation is not suppressed. It is then your responsibility to check each unit or stream that can be OFF. Indeed, calling any function related to an object which is OFF will generate a fatal error.

REPORT PHASE

You can also use FLEX in report phase to design your own reports. The syntax is:

```
FUNCTION REPORT  
...your reporting code ...  
END
```

7.2 USER DEFINED MEASUREMENTS

Flex language can be used to define your own type of measurements on streams. The definition is carried out in two steps:

- defining the measurement type in [ValiModeller](#)
- coding the Flex function describing how the measured property is to be calculated

DEFINING THE MEASUREMENT TYPE IN VALIMODELLER

You must first define the new measurement type, so that ValiModeller can recognize this new type of measurement when creating new tags on streams.

- select the menu command **Options | Configure**
- select the **Model Measurements** tab
- click the **Add** button
- define the measurement properties:

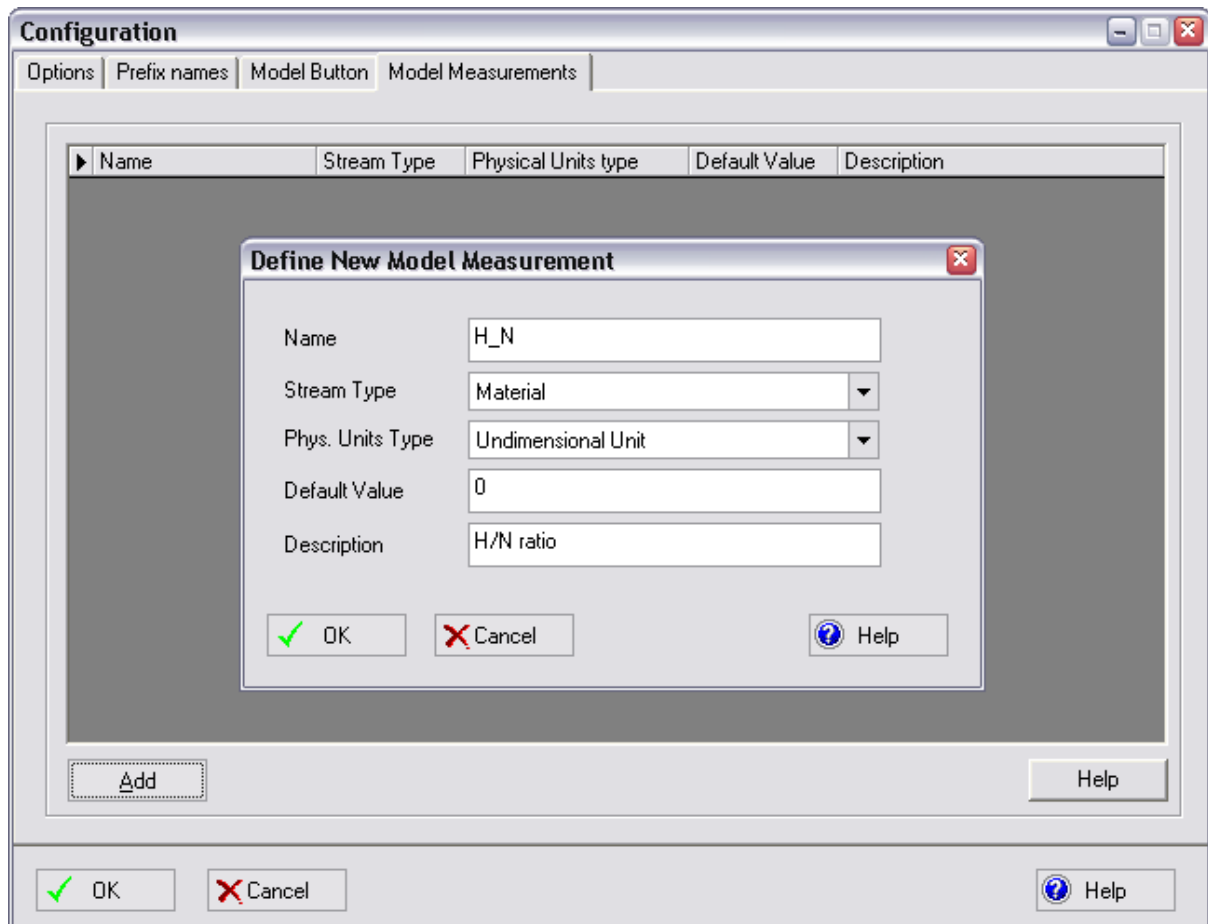


FIGURE - 1 - DEFINING A NEW STREAM MEASUREMENT TYPE

- click twice the **OK** button (once in the **Define New Model Measurement** and once in the **Configuration** window)
- a message window is then displayed mentioning that you must now close the model file and re-open it.
- press again the **OK** button
- click **Yes** to save your modifications
- re-open you model

CODING THE FLEX FUNCTION

In any Flex page (for example in a FLEXLIB type unit), define now how to calculate the property corresponding to this new measurement type.

The result must be stored in the pre-defined variable **RESUSR**. The name of the stream on which the measurement is defined is available in the pre-defined string variable **STRNAM**.

Example:

```
FUNCTION H_N
!=====
! Calculates H/N molar ratio
!=====
CALL GETVAR (STRNAM, 'H2', h2)
CALL GETVAR (STRNAM, 'N2', n2)
RESUSR = h2 / n2
END
```

You can now define a tag of type H/N on any stream that has both H2 and N2 in its corresponding THERMOD.

7.3 FLEX FUNCTIONS TO SELECT VARIABLES AND DATA

7.3.1 GETALL

This function loads the variables of the given STREAM in the arrays STRPAR (stream parameters) and RATESPAR (rates parameters).

The following syntax can be used:

CALL GETALL ('NAME' OR I, 'VARIABLES')

where:

- 'NAME': the name of the stream
- I: number of the stream (positive for inlet stream, negative for outlet stream)
- 'VARIABLES': one of the reserved keywords:
 - 'S': loads only **STRPAR** (stream parameters).
 - 'R': loads only **RATESPAR**
 - 'ALL': loads **STRPAR** and **RATESPAR**

Examples:

```
CALL GETALL('FM','ALL')
! 'FM' is the STREAM inlet name, and must be placed between quotes
! 'ALL': to initialize STRPAR and RATESPAR arrays
```

```
CALL GETALL('FM','S') or CALL GETALL('FM','STREAM')
! 'S' or 'STREAM' to initialise the state variables in STRPAR.
```

```
CALL GETALL(3,'R') or CALL GETALL(3,'RATES')
! 'R' or 'RATES' to initialise RATESPAR from the third inlet stream
```

For non-material streams, the STRPAR array contains the variables of the stream and the RATESPAR is undefined.

For material streams, STRPAR and RATESPAR have the following structure:

- STRPAR array:
 - (1): T: temperature (K)
 - (2): P: pressure (bar)
 - (3): not used (set to 0)
 - (4): RATEFRAC: rates fractions (mixf) (-)
 - (5): H: enthalpy (kJ/kmol)
 - (6): not used (set to 0)

Attention:

1. Only state variables are initialised. For material streams, state variables are usually: T, P, RateFrac. For material streams involving a single compound and of LV, WLV or LLV, the state variables are H, p, RateFrac. For these latter streams, the variable T can however be accessed by using the GETVAR function. For non-material streams, state variables are the stream variables that appear in an equation.

2. The structure of STRPAR allows you to use it directly to call EQPHYS.

- RATESPAR ARRAY: contains the partial flow rates of the STREAM (the others elements of the array are set to zero) in the following form:
 - (1): total molar flow rate (kmol/s)
 - (2): total mass flow rate (kg/s)
 - (3,4,...): partial molar flow rate of COMPOUND 1,2,... (kmol/s)

Attention: The array may be used as an argument for EQPHYS and PROPTY by giving RATESPAR(3) as an argument.

7.3.2 GETVAR

This function retrieves values associated with a STREAM.

The following syntax can be used:

CALL GETVAR ('NAME' OR I, 'VARIABLES', MYVAR)

where:

- 'NAME': the name of the stream
- I: number of the stream (positive for inlet stream, negative for outlet stream)
- 'VARIABLES': one of the reserved keywords:
 - **'STREAM' or 'S'**: copies the state variables of the STREAM into the MYVAR array. For a material STREAM:
 - MYVAR(1): temperature (K)
 - MYVAR(2): pressure (bar)
 - MYVAR(3): not used (set to 0)
 - MYVAR(4): rates fraction (MIXF) (-)
 - MYVAR(5): molar enthalpy (kJ/kmol)
 - MYVAR(6): not used (set to 0)
 - **'RATES' or 'R'**: copies the partial molar flow rates of the stream in the MYVAR array. Its size must be at least the same as the number of components. The MYVAR array may be used as an argument for the EQPHYS and PROPTY functions.
 - 'xxx': where xxx is the variable name:
 - state variable of material stream (xxx = P for pressure, VAPF for the vapour fraction, T for the temperature, H for the enthalpy, MASSF for mass flow rate, MOLF for molar flow rate)
 - name of a compound to get the stream compound partial flow rate
 - any parameter from non material stream
 - any variable of the stream, provided a tag is defined on this variables (e.g. COLDVOLF, COLDDENS, etc.)
- MYVAR: variable to store the result (type REAL)

Examples:

```
CALL GETVAR('FM','H',MYVAR(2))
!'FM' is the Stream Name.
!'H': reserved keyword for a material stream enthalpy.
!MYVAR(2): result array named MYVAR, cell 2.
```

```
CALL GETVAR(-3,'S',MYVAR)
CALL GETALL(-3,'STREAM',MYVAR)
!'to copy the Stream State variables (of the third outlet stream) in MYVAR array.
```

```
CALL GETVAR('FM','R',MYVAR)
CALL GETALL('FM','RATES',MYVAR)
!'to copy the partial molar flow rates in MYVAR array.
```

```
CALL GETVAR('FM','WATER',MYVAR(1))
!'WATER' to copy the partial flow rate of Compound named WATER in MYVAR(1).
```

```
CALL GETVAR('FM','MFWATER',MYVAR(1))
!'WATER' to copy the molar fraction of Compound named WATER in MYVAR(1), assuming such a tag has 1
```

7.3.3 GETUVA

This function retrieves values associated with a UNIT.

The following syntax can be used:

```
CALL GETUVA('NAME' OR ' ', 'VARIABLE', MYVAR)
```

where:

- 'NAME': the name of the stream
- ' ': the unit itself
- 'VARIABLES': one of the UNIT parameters
- MYVAR: variable to store the result (type REAL)

Attention: Only measurable variables of a unit can be accessed through GETUVA. This means that referring to unit parameters such as a MODE type parameter or an Input parameter will not work correctly.

Example

```
CALL GETUVA('SPLIT','DP', dp_split)
!'SPLIT' is the UNIT Name.
!'DP': reserved keyword for pressure drop of the unit.
!dp_split is the local variable
```

7.3.4 GTVARTAG

This function retrieves the validated value associated with a TAG. This is the actual variable associated with the TAG and its value will change between iterations until convergence.

The following syntax can be used:

```
CALL GTVARTAG('NAME',MYVAR)
```

where:

- 'NAME': the name of the TAG
- MYVAR: variable to store the result (type REAL)

Example:

```
CALL GTVARTAG('TI-001',TI)
!'TI-001' is the TAG Name
!TI: result variable
```

7.3.5 GTTAGVAL

This function retrieves the requested value associated with a TAG. This value does not change between iterations.

The following syntax can be used:

CALL GTTAGVAL ('NAME' , 'VALUE' , MYVAR)

where:

- 'NAME': the name of the TAG
- 'VALUE': one of the reserved keywords:
 - **MEA.VAL**: the used value, i.e. value after filtering, compensation and CUTOFF
 - **MEA.ACC**: the used accuracy, i.e. accuracy after filtering, compensation and CUTOFF
 - **SL**: the solver lower bound
 - **LL**: the alarm lower bound
 - **L**: the warning lower bound
 - **H**: the warning upper bound
 - **HH**: the alarm upper bound
 - **SH**: the solver upper bound
 - **DEF.VAL**: the default value, i.e. value entered manually in the model
 - **DEF.ACC**: the default accuracy, i.e. accuracy entered manually in the model
 - **RAW.VAL**: the raw measured value, i.e. value coming from measurement file. If no value in mea file, the default value
 - **RAW.ACC**: the raw measured accuracy, i.e. accuracy coming from measurement file. If no accuracy in mea file, the default accuracy
 - **FILT.VAL**: the filtered value
 - **FILT.ACC**: the filtered accuracy
 - **USED.VAL**: the used value, i.e. value after filtering, compensation and CUTOFF. Same as MEA.VAL
 - **USED.ACC**: the used accuracy, i.e. accuracy after filtering, compensation and CUTOFF. Same as MEA.ACC
 - **IS.OWNER.OFF**: 1 if object (UNIT or STREAM) on which object is defined is OFF, 0 otherwise
 - **IS.OWNER.OUT**: 1 if object (UNIT or STREAM) on which object is defined is OUT, 0 otherwise
 - **IS.BAD**: 1 if tag value is BAD in measurement file, 0 otherwise. Refer to [Bad_data handling](#) for more information
 - **IS.FLAT1**: 1 if tag is FLAT1, 0 otherwise. Refer to [Flat_data detection](#) for more information

- **IS.FLAT2:** 1 if tag is FLAT2, 0 otherwise. Refer to Flat data detection for more information
- **IS.FILTERED:** 1 if tag is filtered by simple filtering (raw value is outside LL HH bounds), 0 otherwise
- **IS.FILTERED.FLEX:** 1 if tag is filtered by FLEX, 0 otherwise
- **MYVAR:** variable to store the result (type REAL)

Example:

```
CALL GTTAGVAL('TI-001','MEA.VAL',TI_MEAS)
!'TI-001' is the TAG Name,
!'MEA.VAL' selects the measured value
!TI_MEAS: result variable.
```

7.3.6 CONVER

This function converts a value from a physical unit to another.

The following syntax can be used:

```
CALL CONVER(VAL1, 'PHYSUN1', VAL2, 'PHYSUN2')
```

where:

- VAL1: value to be converted and expressed in physical units defined by PHYSUN1
- 'PHYSUN1': physical units of the value to be converted (VAL1)
- VAL2: value converted to the physical units defined by PHYSUN2
- 'PHYSUN2': physical units to which the value VAL1 must be converted

7.3.7 ENTRY

This function returns the name of the object stored in location NUM of the dictionary of the model file.

The following syntax can be used:

```
CALL ENTRY(NUM, MYVAR)
```

where:

- NUM: the number of the entry to retrieve
- MYVAR: variable to store the result (type NAME)

7.3.8 GTSTATUS

This function returns the status of a given object (UNIT or STREAM): If the object is ON, the local variable STAT will be set to 1, otherwise it will be set to 0.

The following syntax can be used:

```
CALL GTSTATUS ('NAME', 'TYPE' , MYVAR)
```

where:

- 'NAME': the name of the object
- 'TYPE': the type of object ('U' for UNIT, 'S' for STREAM)
- MYVAR: variable to store the result (type INTEGER)

This function is particularly useful to adapt the formulation of equations depending on the state of some UNIT or STREAM. If you don't use the GTSTATUS function, then the equation will not be generated if any of the objects referred by your code is OFF. This default behaviour can be deactivated by calling the function NOSTACHK.

7.3.9 GETINFO

This function returns the content of any of the INFOi (i=1...9).

The following syntax can be used:

CALL GETINFO ('INFOI', MYVAR)

where:

- 'INFOI': the info to retrieve (i=1...9)
- MYVAR: variable to store the result (type NAME)

Attention: INFOi strings are 80 characters long while NAME variables are only 32 characters long. Only the 32 first characters will be returned by GETINFO.

7.3.10 PUTINFO

This function sets the content of any of the INFOi (i=1...9).

The following syntax can be used:

CALL PUTINFO ('INFOI', MYVAR)

where:

- 'INFOI': the info to set (i=1...9)
- MYVAR: variable that contains the text to set (type NAME). A string can be directly provided

Attention: PUTINFO cannot be used in report phase.

7.3.11 NOSTACHK

This function deactivates the automatic removal of Flex equations which are referencing a unit or stream which is OFF.

The following syntax can be used:

CALL NOSTACHK

7.3.12 REMOVEEQ

This function removes the current equation, and **can only be used for debugging**. This is useful when some equations are linked to a stream or unit which has been set OFF for some reason.

The following syntax can be used:

CALL REMOVEEQ

Attention: when ValiEngine analyzes the Flex Code (function by function, line by line), if it sees **CALL REMOVEEQ**, the equation is removed in any case, even if there is a "IF" before.

7.3.13 LOCATE

2 functions are available to access internal data of an external model file.

Attention: these functions must be manipulated with great care. We recommend that only highly experienced VALI users work with those.

LOCATE1:

This function extracts a scalar value described by 4 input parameters from an external model file and modifies the dictionary of the model file, so that the memory address points to the extracted value.

The following syntax can be used:

CALL LOCATE1 (RESULT, 'INFO', 'TYPE', 'ID', 'NAME')

where

- **RESULT**: the scalar value described by the 4 input parameters
- **'INFO'**: one of the reserved keywords:
 - **'STRUCTUR'** or **'S'**: the list of values ; this keyword can only be used with numerical values of input parameter **'NAME'**.
Note: Using the keyword **'STRUCTUR'** with the function **LOCATE1** has not a lot of sense, because the function only returns a scalar value (see example below). This keyword will be useful with the second function.
 - **'DATA'** or **'D'**: the value
 - **'VAL.DAT'** or **'VAL.D'**: the value after validation
 - **'FO.D'**: the parameter's name
- **'TYPE'**: the type of object (**'U'** for UNIT, **'S'** for STREAM, **'R'** for RATES or **'C'** for COMPOUND)
- **'ID'**: the name of the object
- **'NAME'**: either the name of the parameter (i.e. **'T'** for Temperature, **'P'** for Pressure, **'H'** for Enthalpy,...), or a number corresponding to the index of the parameter in the section of the model file defined by the **'INFO'**, **'TYPE'** and **'ID'** parameters.

Examples:

```
CALL LOCATE1 (TEMP, 'D', 'S', 'STREAM.1', 'T')
TEMP=300
```

In this example, the **LOCATE1** function will not only extract the value of the Temperature parameter of the Stream called "STREAM.1" from a model file, but it will also point to the memory address of the extracted value. As a result, you can replace the Temperature value in the model file using the second command.

```
CALL LOCATE1 (TEMP, 'D', 'S', 'STREAM.1', 2)
```

In this example the **LOCATE1** function will return the value of the second parameter of the Stream called "STREAM.1" (which is the Pressure for a Material Stream for example), together with its memory address.

```
CALL LOCATE1 (TEMP, 'S', 'S', 'STREAM.1', 2)
```

This command is exactly the same than the previous one. In this example the **LOCATE1** function will return the second parameter of the list of values of the Stream called "STREAM.1" (which is the Pressure for a Material Stream for example), together with its memory address.

LOCATEN:

This function extracts a vector of values described by 3 input parameters from an external model file and modifies the dictionary of the model file, so that the memory address points to the extracted vector.

The following syntax can be used:

```
CALL LOCATEN(RESULT, 'INFO', 'TYPE', 'ID')
```

where

- **RESULT**: the vector described by the 3 input parameters
- **'INFO'**: one of the reserved keywords:
 - **'STRUCTUR'** or **'S'**: the list of values ; this keyword can only be used with numerical values of input parameter **'NAME'**.
Attention: When using this keyword, the vector returned by the function LOCATEN has a predefined structure depending on the type of object selected with input parameter 'TYPE' (see details below)
 - **'DATA'** or **'D'**: the value
 - **'VAL.DAT'** or **'VAL.D'**: the value after validation
 - **'FO.D'**: the parameter's name
- **'TYPE'**: the type of object (**'U'** for UNIT, **'S'** for STREAM, **'R'** for RATES or **'C'** for COMPOUND)
- **'ID'**: the name of the object

Example:

```
REAL Struct(60)
```



```
CALL LOCATEN(Struct, 'S', 'U', UniNam)
NFI = Struct(1)

CALL ENTRY(Struct(10+1), StrName)
CALL GETVAR(StrName, 'T', t1)

CALL ENTRY(Struct(NFI+10+1), StrName)
CALL GETVAR(StrName, 'T', t2)
```

When using the LOCATEN function, it is mandatory to declare the output parameter as a vector first. Here, we initialize Struct as a vector of real numbers of length 60.

We use the LOCATEN function to extract the vector of parameters of the UNIT which name is stored in the keyword UniNam, together with their memory address. We define the variable NFI as the first element of the vector Struct, which is the number of inlet streams when using LOCATEN with inputs 'INFO'='S' and 'TYPE'='U'.

We use the ENTRY function to retrieve the name (StrName is the output of the function) of the 10+1 element of the vector Struct, which is the first inlet stream when using LOCATEN with inputs 'INFO'='S' and 'TYPE'='U'. Then, we use the GETVAR function to retrieve the Temperature of this stream.

Finally, we use the ENTRY function to retrieve the name (StrName is the output of the function) of the NFI+10+1 element of the vector Struct, which will be the first outlet stream when using LOCATEN with inputs 'INFO'='S' and 'TYPE'='U'. Then, we use the GETVAR function to retrieve the Temperature of this stream.

Content of the vector returned by the LOCATEN function when using the keyword 'STRUCTUR' or 'S' as 'INFO' input, depending on the 'TYPE' of object:

Notes:

- When the ">" symbol appears in the table below, it means that the vector's cell contains the specified value AND its memory address. Otherwise, it just contains the value.
- For the UNIT vector, the 14th index doesn't always correspond to the number of Reactions, it depends on the number of streams connected to the UNIT. For example, if the UNIT has 2 inlets streams and 3 outlets, indexes 11 to 15 will correspond to the streams, and index 16 will be the number of Reactions.

Index in vector	UNIT	STREAM	RATE (MIXDAT)	COMPOUND
1	Nb of inlet streams	Nb of parameters	Nb of parameters	Nb of parameters
2	Nb of outlet streams	-	-	-
3	Nb of parameters	Nb of parameters	-	> DATA format
4	-	-	-	DATA Rcd

5	Flex flag (=0 if no code, =1 if code)	Nb of parameters	-	Ind BDOTHER
6	> DATA format	> DATA format	> DATA format	Nb Rcd Form
7	DATA Rcd	-	-	Formula Rcd
8	NINF	-	DATA Rcd	Casno
9	INFO Rcd	DATA Rcd	-	Ind CHKTHR
10	FLEX Rcd	> Mixdat	-	Ind CHKTHR
11	> Stream1	TEXT Rcd	> Thermod	Ind CHKTHR
12	> Stream2	-	Attributes AW Rcd	Ind CHKTHR
13	...	DATA Rcd	Attributes W Rcd	-
14	Nb of Reactions	Attributes AW Rcd	Lenght of the attributes record	-
15	> Reaction1	Attributes W Rcd	Nb of used attributes	-
16	> Reaction2	Lenght of the attributes record	Nb of Reactions	-
17	...	Nb of used attributes	Reaction Rcd	-
18	-	> Granul	Nva Reactions	-
19	-	-	TEXT Rcd	-

7.4 FLEX FUNCTIONS FOR THERMODYNAMIC CALCULATIONS

7.4.1 SELECT

This function selects a reference stream defining which methods to use for thermodynamic functions. A reference stream must be defined before functions, like EQPHYS and PROPTY, can be used.

The following syntax can be used:

CALL SELECT ('NAME' OR I)

where:

- 'NAME': the name of the stream
- I: number of the stream (positive for inlet stream, negative for outlet stream)

The following FLEX variables and arrays are initialized when using the SELECT function:

- **STRNAM**: inlet STREAM name (type NAME)
- **STTYPE**: inlet STREAM type (type NAME)
- **STRPARNM**: inlet STREAM parameters names (array) (type NAME(I))
- **RATESNAM**: associated RATES name.(type NAME)
- **RATESTYPE**: associated RATES type (i.e. : MOL or MASS) (type NAME)
- **RATPARNM**: associated RATES names of the components (array) (type NAME(I))
- **NUMSUB**: number of compounds(type REAL)
- **SUBTYPE**: compounds type (i.e. : EPIC or SOLID) (type NAME(I))
- **THERMOD**: thermodynamic model of STREAM (type NAME)

Example:

```
CALL SELECT('AFLX-3')           ! loads stream AFLX-3 as the reference stream.
```

7.4.2 EQPHYS

This function computes phase equilibrium. It is necessary to select a reference stream (with the [SELECT](#) function) previously.

The following syntax can be used:

```
CALL EQPHYS(VEC1, VEC2, VAP, LIQ, SOL, LIQ2, 'TYPE')
```

where:

- VEC1: array with stream variables:
 - VEC1(1): temperature (K)
 - VEC1(2): pressure (bar)
 - VEC1(3): vapour fraction ($VAPF = (V/V+L)$)
 - VEC1(4): not used
 - VEC1(5): enthalpy flow rate (kW)
 - VEC1(6): entropy flow rate (kW/K)
- VEC2: array with partial molar flow rates of each compound. This array may be initialised with a call to GETALL or to GETVAR
- VAP: array with a size of at least the number of compounds of the STREAM, including the partial molar flow rates of the vapour phase after computation (kmol/s)
- LIQ: array with a size of at least the number of compounds of the STREAM, including the partial molar flow rates of the (first) liquid phase after computation (kmol/s)
- SOL: array with a size of at least the number of compounds of the STREAM, including the partial molar flow rates of the solid phase after computation (kmol/s)
- LIQ2: array with a size of at least the number of compounds of the STREAM, including the partial molar flow rates of the (second) liquid phase after computation (kmol/s)
- 'TYPE': a reserved KEYWORD that defines the type of equilibrium (flash) to perform:
 - **EPT**: equilibrium at a given P and T
 - **EPA**: equilibrium at a given P and vapour fraction
 - **ETA**: equilibrium at a given T and vapour fraction
 - **EHP**: equilibrium at a given P and enthalpy
 - **ESP**: equilibrium at a given P and entropy
 - **EPTA**: Extension of EPT with vapour ratio given as an argument but only used in the saturation zone of a single component stream, where T and P do not uniquely define the stream state

The number of phases to be taken into account for the equilibrium computation is set with the variable STTYPE, initialised by SELECT from the type defined for the stream in the model file.

It is possible to change it to compute other flash types: V, L, S, LV, LL, LLV, WL, WLV.

Attention:

1. It is mandatory to give initial values to variables used with the flash type defined; others can be given as initial values.

2. The STRPAR array created by the GETALL (or GETVAR) subroutine has the same structure and may be used as the first argument for EQPHYS.

7.4.3 PROPTY

This function computes physical and thermodynamic properties. It is necessary to select a reference stream (with the [SELECT](#) function) previously.

The following syntax can be used:

```
CALL PROPTY ('FUNCTION', T, P, VEC1, MYVAR)
```

where:

- 'FUNCTION' : one of the reserved keywords that defines the requested computation:
 - **TOTMAS**: total mass flow rate (kg/s)
 - **TOTMOL**: total molar flow rate (kmol/s)
 - **MOLFRA**: molar ratios (mol/mol)
 - **MASFRA**: mass ratios (kg/kg)
 - **MOLDRY**: molar ratios of dry gases
 - **MASFLW**: partial mass flow rates (kg/s)
 - **HFLOW**: enthalpy flow (kW)
 - **HFLOWTOT**: enthalpy flow including upper heating value (kW)
 - **LHVWT**: lower heating value (weight basis) (kJ/kg)
 - **LHVMOL**: lower heating value (molar basis) (kJ/kmol)
 - **HFORM**: standard enthalpy of formation on molar basis (kJ/kmol)
 - **HFORMTOT**: standard enthalpy of formation (kW)
 - for vapour phase properties:
 - **VG**: volumetric flow rate (m³/s)
 - **HG**: enthalpy flow rate (kW)
 - **GG**: free energy flow rate (kW)
 - **SG**: entropy flow rate (kW/K)
 - **CPG**: thermal capacity of stream (kW/K)
 - **CONG**: thermal conductivity (kW/m/K)
 - **VISG**: dynamic viscosity (kg/m/s)

- **SONG**: speed of sound (m/s)
- **FUGG**: fugacity of components (bar)
- for liquid phase properties:
 - **VL**: volumetric flow rate (m³/s)
 - **HL**: enthalpy flow rate (kW)
 - **GL**: free energy flow rate (kW)
 - **SL**: entropy flow rate (kW/K)
 - **CPL**: thermal capacity of stream (kW/K)
 - **CONL**: thermal conductivity (kW/m/K)
 - **VISL**: dynamic viscosity (kg/m/s)
 - **SONL**: speed of sound (m/s)
 - **FUGL**: fugacity of components (bar)
- for solid phase properties:
 - **VS**: volumetric flow rate (m³/s)
 - **HS**: enthalpy flow rate (kW)
 - **GS**: free energy flow rate (kW)
 - **SS**: entropy flow rate (kW/K)
 - **CPS**: thermal capacity of stream (kW/K)
 - **CONS**: thermal conductivity (kW/m/K)
 - **FUGS**: fugacity of components (bar)
- T: temperature (K)
- P: pressure (bar)
- VEC1: array with partial molar flow rates of compounds (kmol/s). When calculating properties for a given phase of a multiphase flow, the array VEC1 must contain the rates of the corresponding phase. These rates can be calculated using the EQPHYS function
- MYVAR: variable to store the result (type REAL). It is an array with a size of at least the number of compounds when requesting for function of composition or fugacity.

7.5 FLEX FUNCTIONS USED IN FILTERING

The following functions are only available in the filtering phase. They cannot be used in the calculation or report phase.

7.5.1 SETOFF

This routine sets an object (UNIT or STREAM) to OFF.

The following syntax can be used:

CALL SETOFF ('NAME', 'TYPE')

where:

- 'NAME': the name of the object to set OFF
- 'TYPE': the type of object ('UNIT' or 'U' for a unit, 'STREAM' or 'S' for a STREAM)

7.5.2 SETAGVAL

This function changes values linked to a tag. The tag will be considered as filtered.

The following syntax can be used:

CALL SETAGVAL ('TAGNAME', 'VALUE', MYVAR OR VALUE)

where:

- 'TAGNAME': the name of the tag
- 'VALUE': one of the reserved key words:
 - **MEA.VAL**: the measured value
 - **MEA.ACC**: the measured accuracy (a negative value can be used, indicating relative uncertainty)
 - **SL**: the solver lower bound
 - **LL**: the alarm lower bound
 - **L**: the warning lower bound
 - **H**: the warning upper bound
 - **HH**: the alarm upper bound
 - **SH**: the solver upper bound
 - **COMPENS**: the compensation factor
 - **BIAS**: the compensation bias
 - **AREF, BREF, CREF**: the compensation parameters Aref, Bref and Cref

7.5.3 PUTAGVAL

This routine works similarly as the [SETAGVAL](#) routine but the tag will not be considered as filtered.

The following syntax can be used:

CALL PUTAGVAL ('TAGNAME', 'VALUE', MYVAR OR VALUE)

where:

- 'TAGNAME': the name of the tag
- 'VALUE': one of the reserved keywords:
 - **MEA.VAL**: the measured value
 - **MEA.ACC**: the measured accuracy (a negative value can be used, indicating relative uncertainty)
 - **SL**: the solver lower bound
 - **LL**: the alarm lower bound
 - **L**: the warning lower bound
 - **H**: the warning upper bound
 - **HH**: the alarm upper bound
 - **SH**: the solver upper bound
 - **COMPENS**: the compensation factor
 - **BIAS**: the compensation bias
 - **AREF, BREF, CREF**: the compensation parameters Aref, Bref and Cref

7.5.4 SETNOEQ

This routine removes the corresponding equation from the Flex code associated to a FLXVAL units.

The following syntax can be used:

CALL SETNOEQ ('NAME', 'EQI')

where:

- 'NAME': the name of the object
- 'EQI': the name of the equation (i=1 ... 9)

Example:

```
FUNCTION FILTER
CALL GTTAGVAL('TI-001','MEA.VAL',T1)
IF ( T1 < 283 ) THEN
  CALL SETOFF('E-1','UNIT')
  CALL SETNOEQ('FLX-1','EQ1')
ELSE
  T2=T1+10
  CALL SETAGVAL('TI-004','MEA.VAL',T2)
  CALL SETAGVAL('TI-004','MEA.ACC',5)
ENDIF
END.
```


SECTION 8: USING FLEX FOR DEFINING KINETICS

The FLEX language can also be used to define the kinetic equation and the reaction enthalpy by using a program linked to the reactor's model. To avoid any interference with other FLEX applications, this kinetic description has to be given in a FLEX function that has the same name as the simulated reactor. This function can appear in after an another FLEX program associated to the reactor.

There are two ways to use FLEX program. It can be used **interpreted** or **compiled**. The choice has to be made in the OPTVAL module. If compiled is chosen it still possible to use interpreted FLEX. For that, the comment **!NOCOMPILE** has to be included just under the instruction **FUNCTION** identifying the kinetic. The compiled FLEX is faster than the interpreted one.

Before the execution of the FLEX program, some variables (like temperature, pressure and composition) are created and initialised. They allow data exchange between the reactor resolution module and the FLEX program that will evaluate the kinetic. The reactions are first calculated using the kinetic parameters defined with the reaction. The FLEX code is then executed, giving the chance to the user to change some or all the reaction velocities.

The following variables are automatically created and are available as input for the calculations:

- TK: absolute temperature (K)
- PBAR: absolute pressure (bar)
- NCOMP: number of compounds
- NREA: number of reactions
- F, Z, PPART, X or C: composition variables (dimension NCOMP); only the variables selected for the Equilibrium and Velocity methods (parameters ETYPE and VTYPE) of the first reaction are available (see also the section "[The Reactions](#)")
- KEQ: equilibrium constants K_i (dimension NREA)
- Kin: kinetic constants K_i' (dimension NREA)

The following variables are automatically created and can be overwritten by the Flex code:

- R: reaction velocities R_i (kmol/m³/s or kmol/kgcata/s) (dimension NREA)
- DHR: reaction enthalpies at the system temperature (kJ/kmol) (dimension NREA)

An example is shown with the RKIVAL example.



belsim

CHAPTER 8

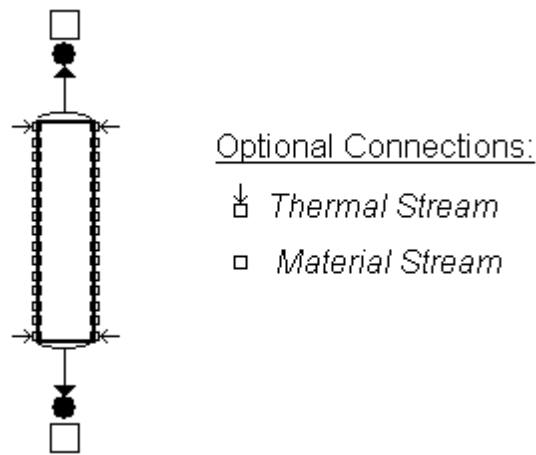
SIMULATION UNITS

SECTION 1: COL010 : DISTILLATION COLUMN

PURPOSE

This is an approximate calculation method for a distillation column based on the Fenske, Underwood and Gilliland equations. This model can be used in simulation or in design mode.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Feed: Material Stream of any state (Liquid, Liquid/Vapour, ...)
	2	Reboiler duty: optional thermal stream
Outlet STREAMS	1	Distillate: Material Stream with LV state
	2	Bottoms: Material Stream with LV state
	3	Condenser duty: optional thermal stream

All material streams must be based on the same Thermod.

UNIT PARAMETERS (SUMMARY)

KPRINT	Input	Print-out Level
MODE	Input	Calculation Mode
COND	Input	Condenser Type
KEYi	Input	Key Compounds (i=1,2)
FRACi	Measurable	Key Compound Fractions (can be manipulated by a controller)
REFLRAT	Input/Result	Reflux Ratio
RSRMIN	Input	Actual/Min Reflux ratioC
THPLAT	Input/Result	Theoretical plates Number
MINPLAT	Result	Min. plates # (Total reflux)
FEEDPLAT	Input/Result	Feed plate
CONDLOAD	Result	Condenser Load
REBLOAD	Result	Reboiler Load

UNIT PARAMETERS

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

MODE	Input	Calculation Mode

- 1 Simulation mode.
- 2 Design by the equations of Fenske, Underwood and Gilliland, feed plate chosen by the Fenske equation.
- 3 Design by the equations of Fenske, Underwood and Gilliland, feed plate chosen by the Kirbride equation.
- 4 Design by the equations of Fenske, Hengstebeck and McCabe-Thiele.

COND	Input	Condenser Type

- 0 Total condenser.
- 1 Partial condenser.

KEYi	Input	Key Compounds (i=1,2)
FRACi	Measurable	Key Compound Fractions (can be manipulated by a controller)

These parameters are displayed in the Others page. They indicate the fractions of the key compounds

that are recovered in the distillat.

In simulation mode only the FRAC1 parameter is used as input.

REFLRAT Input/Result Reflux Ratio

The reflux ratio is defined as the ratio between the reflux and the distillate molar flow rates.

RSRMIN Input Actual/Min Reflux ratioC

This is the ratio between the actual and the minimum reflux ratio

THPLAT Input/Result Theoretical plates Number

The number of theoretical is used as input in simulation mode. It will be calculated in the design modes

MINPLAT Result Min. plates # (Total reflux)

This parameter is the minimum number of plates (at total reflux)

FEEDPLAT Input/Result Feed plate

The feed plate number is used as input in simulation mode. It will be calculated in the design modes

CONDLOAD Result Condenser Load

REBLOAD Result Reboiler Load

SECTION 2: CTL010 : SINGLE VARIABLE CONTROLLER

PURPOSE

The **CTL010** unit is a controller involving one controlled variable and one manipulated variable. These variables must be defined as Tags. For the controlled variable, the measured value of the Tag will be used as set points.

For control loops involving several variables, you should use units of type CTL020 or OPT070.

Please note that, as any other convergence unit, the **CTL010** unit should be the last unit of the sequence defined by a Simulation PFD.

SYMBOLS AND CONNECTION RULES



CTL010

You cannot connect any stream to a **CTL010** unit.

UNIT PARAMETERS (SUMMARY)

FLASH	Input	Stream Initialisation Mode
KPRINT	Input	Print-out Level
ITMAX	Input	Maximum number of iterations
EPSX	Input	Precision asked on the manipulated variable
EPSF	Input	Precision asked on the controlled variable
TEST	Input	Threshold value between relative and absolute precision tests on variables
DXMAX	Input	Maximum variation (absolute value) of the variable at an iteration
DX0	Input	Initial variation of the variable (for 1st iteration)
DFMAX	Input	Maximum ratio of the functions at two points for the secant method
DF0	Input	Ratio used when the function ratio at two points is higher than DFMAX

DEFINING THE CONTROL LOOP

The controlled and manipulated variables are defined via the **Control Tags** page of the Unit Notebook. The list of existing Tags is shown in the **Tags Defined** list. You can use the **Mask** field to limit the number of tags displayed in this list. A simple control loops involves two variables:

- the **controlled** variable, for which a set-point is specified via the Tag associated to the variable. The following types of Tags are supported:

Material Streams	T, P, VAPF, MIXF, H, MOLF, MASSF, VOLF, Partial Flowrates, MF, WF, MG, WG
Thermal Streams	LOAD
Mechanical Streams	POWER
Units	Any measurable output parameter

- the **manipulated** variable which will be tuned in such a way that the controlled variable reaches the given set-point. The measured value associated to the TAG defining the manipulated variable is not used, but its safety bounds (SL and SH) will be used as lower and upper bounds. The following types of Tags are supported:

Material Streams	T, P, VAPF, MIXF, H, MOLF, MASSF, Partial Flowrates
Thermal Streams	LOAD
Mechanical Streams	POWER
Units	Any measurable output parameter

Note: Some of the unit parameters can be either input (given) or output (result) depending on their **MODE of operation**.

UNIT PARAMETERS

FLASH Input Stream Initialisation Mode

When the manipulated variable belongs to a material stream, the parameter FLASH indicates how to reinitialise the stream:

- 1 Stream state is not calculated
- 0 Material balance only; the stream enthalpy is set to -1
- 1 Stream state is based on its pressure and temperature (EPT)
- 2 Stream state is based on its pressure and vapour fraction (EPA)
- 3 Stream state is based on its temperature and vapour fraction (ETA)
- 4 Stream state is based on its enthalpy and pressure (EPH)
- 5 Stream state is calculated according to its own initialisation mode

KPRINT Input Print-out Level

Controls the amount of information printed in the trace file.

ITMAX	Input	Maximum number of iterations

Maximum number of iterations.

If ITMAX = 0 then ITMAX is set to 30.

If ITMAX > 0 then ITMAX is the maximum number of iterations : The simulation is stopped if convergence is not achieved within ITMAX iterations.

If $ITMAX < 0$ The simulation proceeds either at convergence, or after $|ITMAX|$ iterations.

EPSX	Input	Precision asked on the manipulated variable

If EPSX ≤ 0 , default value is 1.D-04

EPSF	Input	Precision asked on the controlled variable

If EPSF <= 0 , default value is 1.D-05

TEST	Input	Threshold value between relative and absolute precision tests on variables

If TEST < 0, default value = .01

DXMAX	Input	Maximum variation (absolute value) of the variable at an iteration

If DXMAX ≤ 0 , a default value=.75*(TEST+X).

DX0	Input	Initial variation of the variable (for 1st iteration)

If $DX0 = 0$, a default value = $.25 \cdot (\text{TEST} + X)$.

Suggested values : molar flowrate: -0.15
 pressure: 0.02

DFMAX	Input	Maximum ratio of the functions at two points for the secant method

If DFMAX ≤ 0 , a default value is considered.

DF0	Input	Ratio used when the function ratio at two points is higher than DEMAX

If $DF0 \leq 0$, a default value is considered.

SECTION 3: CTL020 : MULTI-VARIABLES CONTROLLER

PURPOSE

The **CTL020** unit is a controller involving up to five controlled variables and the same number of manipulated variable. These variables must be defined as Tags. For the controlled variables, the measured values of the Tag will be used as set points.

For single variable control loops, you should use units of type CTL010.

Please note that, as any other convergence unit, the **CTL020** unit should be the last unit of the sequence defined by a Simulation PFD.

SYMBOLS AND CONNECTION RULES



CTL020

You cannot connect any stream to a **CTL020** unit.

UNIT PARAMETERS (SUMMARY)

KPRINT	Input	Print-out Level
ITMAX	Input	Maximum number of iterations
NVAR	Input	Number of variables
DX0_i	Input	Value of the disturbance of the variable (<i>i</i>=1,5)
FLASH_i	Input	Stream Initialisation Mode (<i>i</i>=1,5)
EPS_i	Input	Precision required for the controlled variable (<i>i</i>=1,5)

DEFINING THE CONTROL LOOP

The controlled and manipulated variables are defined via the **Control Tags** page of the Unit Notebook. The list of existing Tags is shown in the **Tags Defined** list. You can use the **Mask** field to limit the number of tags displayed in this list. A simple control loops involves two variables:

- the **controlled** variable, for which a set-point is specified via the Tag associated to the variable. The following types of Tags are supported:

Material Streams	T, P, VAPF, MIXF, H, MOLF, MASSF, VOLF, Partial Flowrates, MF, WF, MG, WG
Thermal Streams	LOAD
Mechanical Streams	POWER
Units	Any measurable output parameter

- the **manipulated** variable which will be tuned in such a way that the controlled variable reaches the given set-point. The measured value associated to the TAG defining the manipulated variable is not used, but its safety bounds (SL and SH) will be used as lower and upper bounds. The following types of Tags are supported:

Material Streams	T, P, VAPF, MIXF, H, MOLF, MASSF, Partial Flowrates
Thermal Streams	LOAD
Mechanical Streams	POWER
Units	Any measurable output parameter

Note: Some of the unit parameters can be either input (given) or output (result) depending on their **MODE of operation**.

UNIT PARAMETERS

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

ITMAX **Input** **Maximum number of iterations**

ITMAX = 0 then ITMAX is set to 20.

ITMAX > 0 The simulation is stopped if convergence is not achieved within ITMAX iterations.

ITMAX < 0 The simulation proceeds either at convergence, or after |ITMAX| iterations.

NVAR **Input** **Number of variables**

Specifies the number of variables actually controlled. If more tags are defined in the Control Tags page, they will be ignored.

DX0i	Input	Value of the disturbance of the variable ($i=1,5$)

If $DX0i < 0$ $|DX0i|$ is the fraction by which the variable will be varied for calculating the derivative

If $DX0i > 0$ $|DX0i|$ is the value by which the variable will be varied for calculating the derivative

FLASHi	Input	Stream Initialisation Mode ($i=1,5$)

When the manipulated variable belongs to a material stream, the parameter FLASHi indicates how to reinitialise the stream:

- 1 Stream state is not calculated
- 0 Material balance only; the stream enthalpy is set to -1
- 1 Stream state is based on its pressure and temperature (EPT)
- 2 Stream state is based on its pressure and vapour fraction (EPA)
- 3 Stream state is based on its temperature and vapour fraction (ETA)
- 4 Stream state is based on its enthalpy and pressure (EPH)

EPSi	Input	Precision required for the controlled variable ($i=1,5$)

EPSi = 0 default value is 1.D-04

EPSi < 0 the precision is given on a relative basis (fraction)

EPSi > 0 EPSi is the precision on an absolute basis

SECTION 4: CUT010 : UTILITY TO CUT STREAMS

PURPOSE

Utility that handles the cuts for OPT070 convergence promoter.

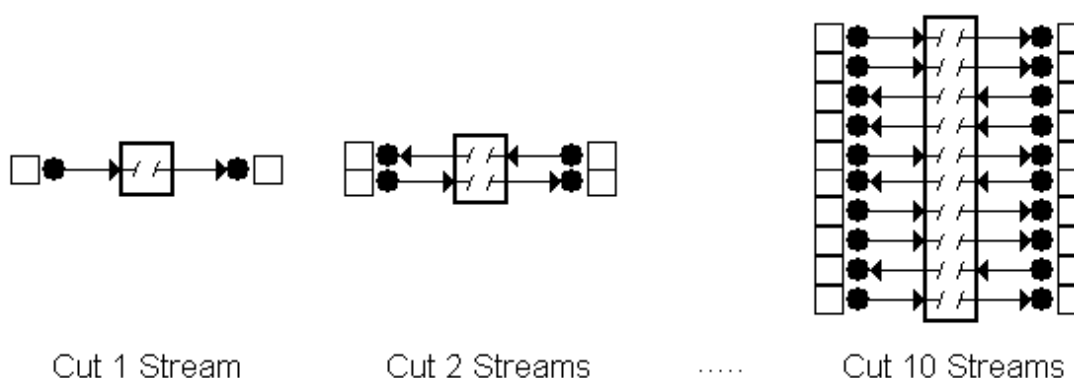
Up to 10 streams can be handled simultaneously.

Convergence promotion on molar enthalpy is preferred to promotion on temperature, so as to avoid problems in case of a phase change in a single component mixture (pure compound). Estimated values for the torn variables are provided in the outlet STREAMS according to the situation. The process is then calculated to generate process answers that are available in the inlet STREAMS.

The **CUT010** UNIT must be the last UNIT in a Simulation PFD. This simulation PFD must be a sub-PFD of the PFD containing the OPT070 optimiser unit.

The maximum number of torn variables is equal to 100.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS Maximum number : 10.
Any type may be used.

Outlet STREAMS Maximum number : 10.
Any type may be used.

Corresponding inlet and outlet streams must be of the same type. In addition, corresponding material streams must use the same Thermod.

UNIT PARAMETERS (SUMMARY)

KPRINT	Input	Print-out Level
TEARTY <i>i</i>	Input	Tear type for inlet stream <i>i</i>

UNIT PARAMETERS

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

TEARTY <i>i</i>	Input	Tear type for inlet stream <i>i</i>

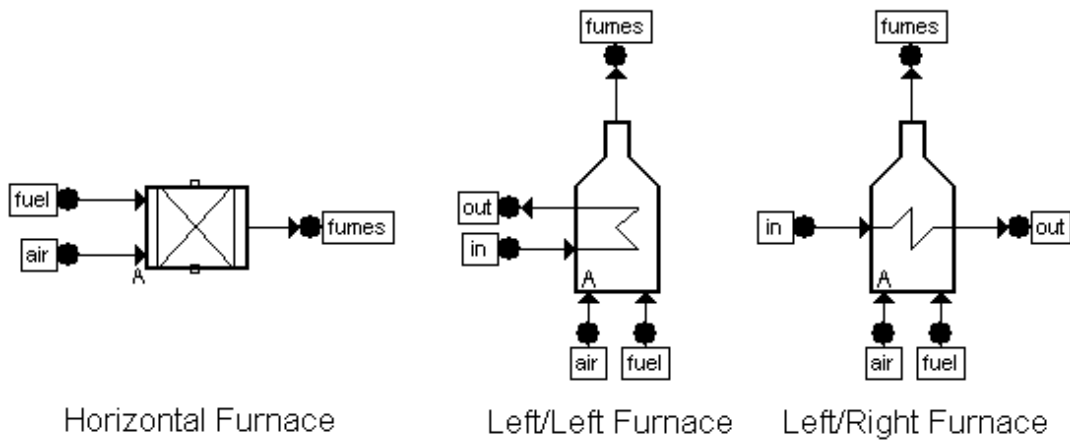
- 1 Only the total molar flowrate is torn.
- 2 Only the molar enthalpy is torn.
- 3 The total molar flowrate and molar enthalpy are torn.
- 4 The molar compositions are torn.
- 5 The partial molar flow rates and the molar enthalpy are torn.
- 6 The partial molar flow rates and the temperature are torn.
- 7 The partial molar flow rates are torn.
- 8 LOAD
- 9 POWER

SECTION 5: FUR010 : FURNACE

PURPOSE

FUR010 calculates the heat and mass balances of a burner. Optionally a stream can be heated up by the fumes. Various modes are available to determine the air flow rate.

SYMBOLS AND CONNECTION RULES



Optional Connection:

- Material Stream

Inlet STREAMS	1	Air Feed: material stream bringing the oxygen
	2	Fuel Feed: material stream bringing the fuel
	3	optional material stream to be heated up by the fumes
Outlet STREAMS	1	Fumes: material stream
	2	optional material stream heated up by the fumes

UNIT PARAMETERS (SUMMARY)

MODEXC	Input	Air Flow Calculation Mode
MODECAL	Input	Calculation Mode
KPRINT	Input	Print-out Level
SETPOINT	Measurable	Air excess or fumes O₂/CO₂ contents(can be manipulated)
TFUMES	Measurable	Outlet temperature of the fumes (can be manipulated)
TOUT	Measurable	Outlet temperature of the fluid (can be manipulated)
ALFOUT	Measurable	Fluid outlet vapour fraction (can be manipulated)

UNIT PARAMETERS

MODEXC	Input	Air Flow Calculation Mode

- 0 The air flow rate is given in the Air inlet stream
- 1 SETPOINT on the air excess
- 2 SETPOINT on the O₂ content in fumes
- 3 SETPOINT on the CO₂ content in dry fumes
- 4 SETPOINT on the CO₂ content in fumes

For MODEXC = 1..4, the flow rate of the Air inlet stream will be calculated.

MODECAL	Input	Calculation Mode

BURNER MODES

MODECAL = 0	The fuel and air compositions, their temperatures and the fuel flowrates are given. The adiabatic temperature and the composition of the fumes are calculated according to the specified Air Flowrate Calculation Mode.
MODECAL = 3	Mass balance mode. The fuel and air composition, and the fuel flow rate are given. The composition of the fumes is calculated according to the specified Air Flowrate Calculation Mode.

FURNACE MODES

MODECAL = 1	The fuel and air compositions, and their temperatures are given. The flowrate of the stream to be heated is also given along with its inlet and outlet temperatures. The fumes outlet temperature is specified. The fumes composition and fuel flowrate, are calculated according to the specified Air Flowrate Calculation Mode.
MODECAL = 2	The fuel and air compositions, and their temperatures are given. The flowrate of the stream to be heated is also given along with its inlet temperature and its outlet vapour fraction. The fumes outlet temperature is specified. The fumes composition and fuel flowrate are calculated according to the specified Air Flowrate Calculation Mode.

KPRINT Input Print-out Level

Controls the amount of information printed in the trace file.

SETPOINT Measurable Air excess or fumes O₂/CO₂ contents(can be manipulated)

The SETPOINT is used for controlling the air flow rate, according to the Air Flow Rate Calculation Mode defined by the MODEXC parameter. When MODEXC = 0, the air flowrate is given in the air Stream.

TFUMES Measurable Outlet temperature of the fumes (can be manipulated)

Outlet temperature of the fumes (MODCAL=1,2)

TOUT Measurable Outlet temperature of the fluid (can be manipulated)

Outlet temperature of the fluid (MODCAL=1)

ALFOUT Measurable Fluid outlet vapour fraction (can be manipulated)

Fluid outlet vapour fraction (MODCAL=2), the state corresponds to the bubble point or dew point respectively if ALFOUT is equal to 0 or 1.

SECTION 6: HTX010 : HEAT TRANSFER

PURPOSE

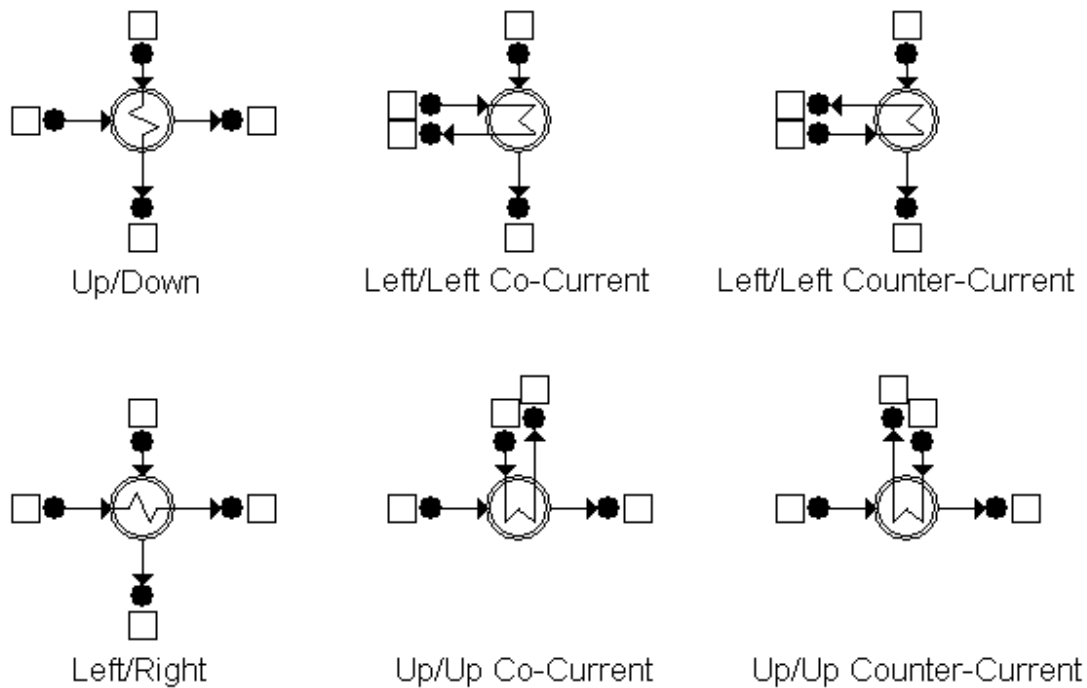
Calculates the energy balance and the heat transfer of a heat exchanger using a global HA factor:

$$\text{LOAD} = \text{HA} \cdot \text{DTLM}$$

Where: DTLM is the logarithmic average temperature difference between the two streams.

There are two inlet streams and two outlet streams. The temperatures, pressures and compositions of both inlets are known. Various design and simulation modes are available. In simulation mode, the heat transfer coefficient and the pressure drops can be made linked to the flow rates.

SYMBOLS AND CONNECTION RULES



Inlet streams 2 Material streams (any state)

Outlet streams 2 Material streams (any state)

Corresponding inlet and outlet streams must use the same Thermod

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation Mode
KPRINT	Input	Print-out Level
HA0	Input	Tube resistance (Conduction coefficient)
EPST	Input	Precision on T
EPSHA	Input	Precision on HA
ITMAX	Input	Maximum number of iterations
HACALC	Result	Overall heat transfer
LOAD	Result	Computed Heat Load
DP _{<i>i</i>}	Measurable	Reference pressure drop for side <i>i</i>
DPMAS <i>i</i>	Input	Reference mass flow rate for side <i>i</i>
DPEXP <i>i</i>	Input	Exponent for flow rate dependence in side <i>i</i>
DPRO <i>i</i>	Input	Reference density for side <i>i</i>
HA _{<i>i</i>}	Input	Reference heat transfer coefficient for side <i>i</i>
HMASS <i>i</i>	Input	Reference mass flow rate for side <i>i</i>
HEXP <i>i</i>	Input	Exponent for flow rate dependence in side <i>i</i>
TOUT <i>i</i>	Measurable	Outlet temperature of stream <i>i</i>
ALFOUT <i>i</i>	Measurable	Outlet vapour fraction of stream <i>i</i>

UNIT PARAMETERS

MODE	Input	Calculation Mode

In **design** modes, the Heat Transfer Factor HA is calculated in order to match various set-points:

The flow rate of the first inlet stream is always kept.

MODE=1	Specified: Flow2, Tout1	Calculated: Tout2, HA
MODE=2	Specified: Flow2, ALFout1	Calculated: Tout2, HA
MODE=7	Specified: Tout1, ALFout2	Calculated: Flow2, HA
MODE=8	Specified: Tout1, Tout2	Calculated: Flow2, HA
MODE=9	Specified: Flow2, ALFout1	Calculated: Tout2, HA
MODE=10	Specified: ALFout1, ALFout2	Calculated: Flow2, HA
MODE=11	Specified: ALFout1, Tout2	Calculated: Flow2, HA

In **simulation** mode, the Heat Transfer Factor HA is calculated on basis of the following formula:

$$1 \quad 1 \quad 1 \quad 1$$

----- = ----- + ----- + -----

HACALC HA0 HA_1 (MASSF1/HMASS1)HEXP1 HA_2 (MASSF2/HMASS2)HEXP2

In all cases, except MODE=0 (true simulation), the flow rate of the second stream will be adjusted in order to match various conditions on the first or second outlet stream state.

MODE=0	Specified: HA, Flow2	Calculated: Tout1, Tout2
MODE=3	Specified: HA, Tout1	Calculated: Flow2, Tout2
MODE=4	Specified: HA, ALFout1	Calculated: Flow2, Tout2
MODE=5	Specified: HA, Tout2	Calculated: Flow2, Tout1
MODE=6	Specified: HA, ALFout2	Calculated: Flow2, Tout1

KPRINT Input Print-out Level

Controls the amount of information printed in the trace file.

HA0 Input Tube resistance (Conduction coefficient)

This parameter is used as input in simulation mode

If HA0 <= 0, the tube thermal resistance is neglected

EPST Input Precision on T

Relative precision on temperature.

If EPST <= 0, the default value 1.D-04 is used.

EPSHA Input Precision on HA

Relative precision on HA.

If EPSHA <= 0, the default value 1.D-04 is used.

ITMAX Input Maximum number of iterations

If ITMAX <= 0, the default value 30 is used.

HACALC Result Overall heat transfer

In design mode, this parameter is adjusted to get the required stream outlet conditions.

In simulation mode, this parameter is calculated on basis of the various thermal resistances.

LOAD	Result	Computed Heat Load

This parameter is always calculated.

DP_{<i>i</i>}	Measurable	Reference pressure drop for side <i>i</i>
DPMAS<i>i</i>	Input	Reference mass flow rate for side <i>i</i>
DPEXP<i>i</i>	Input	Exponent for flow rate dependence in side <i>i</i>
DPRO<i>i</i>	Input	Reference density for side <i>i</i>

These parameters are used in all modes to calculate the pressure drop on each side, according to the following rules:

If **DP_{*i*} < 0** the outlet stream pressure is set to |DP_{*i*}|
 If **DP_{*i*} = 0** the outlet stream pressure is equal to the inlet pressure
 If **DP_{*i*} > 0** the outlet stream pressure is calculated as:

$$POUT_i = PIN_i - DP_i (MASS_i / DPMAS_i) DPEXP_i (DPRO_i / RO_i)$$

HA_{<i>i</i>}	Input	Reference heat transfer coefficient for side <i>i</i>
HMASS<i>i</i>	Input	Reference mass flow rate for side <i>i</i>
HEXP<i>i</i>	Input	Exponent for flow rate dependence in side <i>i</i>

These parameters are used in the simulation modes to calculate the heat transfer factor.

TOUT<i>i</i>	Measurable	Outlet temperature of stream <i>i</i>
ALFOUT<i>i</i>	Measurable	Outlet vapour fraction of stream <i>i</i>

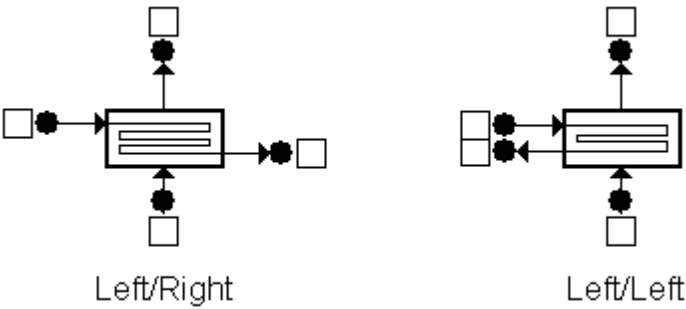
Depending on the calculation **MODE**, these parameters will be used to specify the required conditions for the outlet streams.

SECTION 7: HTX040 : HEAT EXCHANGER

PURPOSE

The HTX040 UNIT simulates a heat exchanger where fumes going through the heating flue are heating a gas or a liquid flowing through tubes (co-current or counter current flow exchanger). Vaporisation can be performed in the tubes. These can either be smooth or finned tubes.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Fumes: material stream (V or LV state)
	2	Fluid to be heated up (any state)
Outlet STREAMS	1	Fumes: material stream (LV)
	2	Heated up fluid (LV)

Corresponding inlet and outlet streams must use the same Thermod.

UNIT PARAMETERS (SUMMARY)

BANKTYP	Input	Bundle Type
HTXTYP	Input	Exchange type
FINTYPE	Input	Type of Fins
NTUBES	Input	Number of Tubes
NTUBPASS	Input	Number of Tubes Passes
LENGTH	Input	Layer Length
DEXT	Input	Tube Outside Diameter
DIN	Input	Tube Inside Diameter
CONDTUB	Input	Thermal Conductivity of Tubes
HPITCH	Input	Horizontal Pitch
KPRINT	Input	Print-out Level
FT	Input	Fin Thickness
NFM	Input	Number of Fins Per Unit Length
FD	Input	Fin External Diameter
CFH	Input	Correction Factor For Heat Transfer Coefficient
CFDPFLUE	Input	Correction Factor For Pressure Drop on Fumes Side
CFDPTUBE	Input	Correction Factor For Pressure Drop in Tubes
NLAYER	Input	Number of Layers
EPS	Input	Relative Precision

UNIT PARAMETERS

BANKTYP	Input	Bundle Type
1		Smooth tubes set out in a row
2		Smooth tubes set out in alternate rows
3		Finned tubes set out in a row
4		Finned tubes set out in alternate rows
HTXTYP	Input	Exchange type
1		Co-Current Flow Exchanger
2		Counter-Current Flow Exchanger

FINTYPE **Input** **Type of Fins**

- 1 Fins of Constant Thickness
- 2 Fins of Variable Thickness

NTUBES **Input** **Number of Tubes**

Total amount of tubes.

Tubes are assembled in parallel planes called LAYERS.

NTUBPASS **Input** **Number of Tubes Passes**

The ratio NTUBES/NTUBPASS must be an integer.

The number of tube passes is equal to 1 if the tubes are independent. When the tubes go through different layers through connections between layers, then the number of tube passes will then be equal to the number of connections + 1.

LENGTH **Input** **Layer Length**

DEXT **Input** **Tube Outside Diameter**

DIN **Input** **Tube Inside Diameter**

Attention:

1. **The number of inlet tubes in the exchanger = NTUBES / NTUBPASS**
2. **The total length of a tube is equal to LENGTH * NTUBPASS**
3. **The total exchange surface area is equal to PI * DEXT * LENGTH * NTUBES.**

CONDTUB **Input** **Thermal Conductivity of Tubes**

If CONDTUB <= 0, then the tubes thermal resistance is neglected.

HPITCH **Input** **Horizontal Pitch**

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

FT	Input	Fin Thickness
NFM	Input	Number of Fins Per Unit Length
FD	Input	Fin External Diameter
CFH	Input	Correction Factor For Heat Transfer Coefficient
CFDPFLUE	Input	Correction Factor For Pressure Drop on Fumes Side
CFDPTUBE	Input	Correction Factor For Pressure Drop in Tubes

These parameters allow tuning the correlations so as to match the actual operation of the equipment.

If a negative value is entered, then the default value 1.0 will be used.

NLAYER	Input	Number of Layers

The number of layers must always be such that the ratio NLAYER/NTUBPASS is an integer value.

NLAYER	Input	Number of Layers

Relative precision required at convergence between the actual exchange surface and the calculated surface.

If EPS $\leq$ 0, the default value 1.E-3 is used.

SECTION 8: HTX050 : FURNACE TO HEAT UP A GAS

PURPOSE

The **HTX050** UNIT is a furnace simulation model used to heat up a gas.

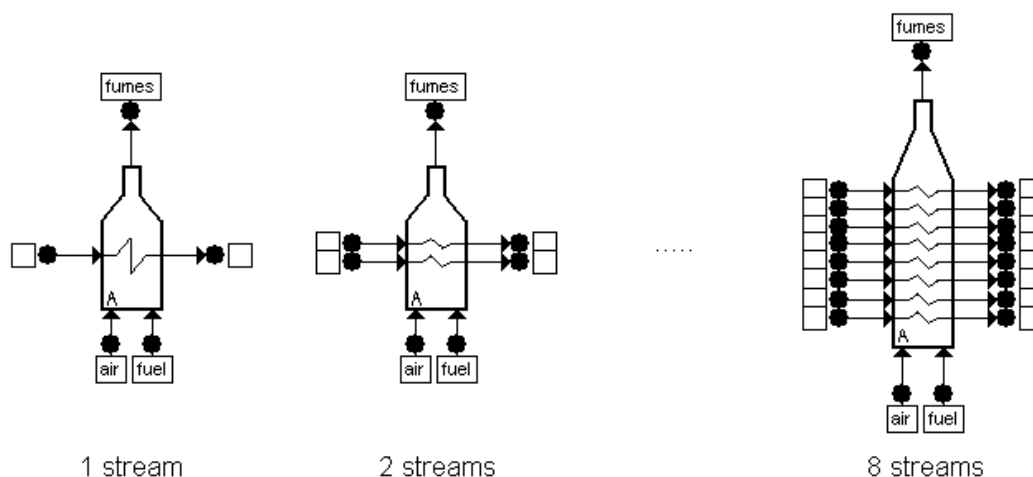
The fuel is liquid or gaseous and its combustion is complete.

The necessary air flowrate is calculated according to the calculation mode for the excess air (MODEXC).

Fuel combustion is described by chemical reactions.

The fuel is assumed to be fully burned.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS

- A minimum of 3 inlet streams is required (maximum 10).
- Their type must be MAT.
- Any STREAM State may be used.

- (1) : Air feed.
 (2) : Fuel.
 (3) to (10) : Gas(es) to be heated up.

Outlet STREAMS

- A minimum of 2 outlet streams is required (maximum 9).
- Their type must be MAT.
- Any STREAM State may be used.

- (1) : Fumes
 (2) to (9) : Heated gas.

The THERMOD associated with the fumes must be made up of all the Compounds which are present in the air as well as in the fuel.

The THERMOD associated to the nth inlet STREAM must be identical to the one associated to the (n-1)th outlet STREAM.

UNIT PARAMETERS

LTUB	Input	TUBE LENGTH
DEXT	Input	TUBE OUTSIDE DIAMETER
DINT	Input	TUBE INSIDE DIAMETER
COND	Input	THERMAL CONDUCTIVITY

If COND ≤ 0, then the tubes thermal resistance is ignored.

FCHINT	Input	CORREC. FACTOR FOR TRANSFER COEFFICIENT
MODEXC	Input	AIR Flow calculation mode

AIREXC	-	Excess of air in relation to neutral combustion.
O2	-	O2 content in the fumes.
MGO2	-	O2 content in the fumes on dry-basis.
CO2	-	CO2 content in the fumes.
MGCO2	-	CO2 content in the fumes on dry-basis.

SETPOINT	Input	SETPOINT FOR AIR EXCESS
MODECAL	Input	COMPUTATION MODE

Since the objective is to determine :

- The outlet temperature of the gas produced by a fixed fuel feed
- or the fuel feed required to heat a gas at a given temperature,

two calculation (Computation) modes are available :

MODECAL = 1	-	Set fuel flow rate.
MODECAL = 2	-	Set one gas outlet temperature.

PARCAL1	Input	Order of the outlet stream

if MODECAL = 2

PARCAL2	Input	Outlet temperature

if MODECAL = 2

GEOMFACT	Input	GEOMETRIC FACTOR FOR HEAT TRANSFER COEFFICIENT

The heat exchange occurs by radiation and depends on the furnace type which is expressed by the GEOMFACT parameter defined by the equation :

$$Q = \text{GEOMFACT} \cdot \text{SIGMA} \cdot \text{AREA} \cdot (\text{TFUMES}^4 \cdot \text{TWALL}^4)$$

$$\text{SIGMA} = 5.667\text{E-}11 \quad (\text{kJ/S/m}^2/\text{K}^4)$$

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

EPS	Input	RELATIVE PRECISION

Relative precision required with regard to thermal load of exchanger.

If $\text{EPS} \leq 0$, $\text{EPS} = 1.\text{D-}04$ is taken.

DATA FOR THE DIFFERENT GASES TO BE HEATED TABLE :

Tubes	-	Total number of tubes for input STREAM i+2.
Feed	-	Number of inlet tubes for input i+2.
Pres. Drop	-	Reference pressure drop for input stream (i+2).
DPi<0	-	
DPi=0	-	no pressure drop
DPi<0	-	$P_{outi} = P_{ini} - DP_i * (MASS_{Fi}/DPMASS_i)^2 * DPRO_i/RO_i$
Mass Flow	-	Reference mass flow rate for pressure drop (kg/s).
If DMASSi = 0	-	no flowrate dependency.
If DMASSi < 0	-	first value of mass flowrate will be saved as reference value for next simulation.
Specif. Mass	-	Reference specific mass (kg/m ³) for Pouti.
If DROi = 0	-	no density dependency.
If DROi < 0	-	the first value of specific mass will be saved as reference value for the next simulation.
Wall Temp.	-	Tube wall temperature for input STREAM i+2.
Heat Load	-	Heat load given to input STREAM i+2.

SECTION 9: HTX060 : BOILER WITH ASSOCIATED DRUM

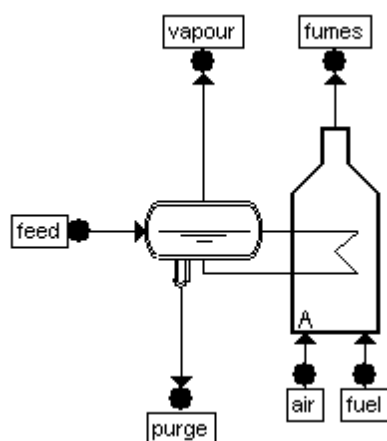
PURPOSE

The **HTX060** UNIT is simulation model for a boiler with an associated drum.

The fuel is liquid or gaseous and its combustion, as defined in the chemical reactions, is complete. The necessary air flow rate is calculated according to the calculation mode of the excess of air (MODEXC).

The vapour fraction at the outlet of the vaporiser loop or the circulation flow rate in the vaporiser loop is a simulation parameter to be set. The heat transfer is calculated on basis of radiation between the fumes and the tubes.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Air Feed: material stream bringing the oxygen (any state)
	2	Fuel Feed: material stream bringing the fuel (any state)
	3	Feed Water to be vaporised

Outlet STREAMS	1	Fumes: material stream
	2	Saturated vapour stream: material stream
	3	Purge: material stream

The THERMOD associated with the fumes must, at least, contain all compounds that are present in the air as well as in the fuel.

The water feed stream, the saturated vapour stream and the purge must use the same THERMOD and this THERMOD must contain only one compound.

UNIT PARAMETERS (SUMMARY)

MODEXC	Input	Air flow calculation mode
MODECAL	Input	Computation mode
NTUBES	Input	Number of tubes
NPASSES	Input	Number of passes
LENGTH	Input	Tubes length
DEXT	Input	Tube outside diameter
DINT	Input	Tube inside diameter
COND	Input	Tube Thermal conductivity
GEOMFACT	Input	Geometric factor for heat transfer coefficient
KPRINT	Input	Print-out Level
STEAMP	Input	Drum pressure (can be manipulated)
CIRCFLOW	Input	Flow rate in the vaporiser (can be manipulated)
ALF	Input	Outlet vapour fraction (can be manipulated)
SETPOINT	Input	Air excess or fumes O ₂ /CO ₂ contents (can be manipulated)
PURGE	Input	Purge as a fraction of feed water (can be manipulated)
CFH	Input	Heat transfer correction factor (can be manipulated)
EPS	Input	Relative precision for thermal balance

UNIT PARAMETERS

MODEXC Input Air flow calculation mode

- 1 Air excess (vs. stoichiometric combustion)
- 2 O₂ content in the fumes
- 3 CO₂ content in the fumes
- 4 CO₂ content in the fumes on dry-basis

MODECAL Input Computation mode

- 1 Fixed feed water flow rate
- 2 Fixed fuel flow rate

NTUBES	Input	Number of tubes
NPASSES	Input	Number of passes
LENGTH	Input	Tubes length
DEXT	Input	Tube outside diameter
DINT	Input	Tube inside diameter

Note: The total exchange surface AREA is equal to $\text{PI} * \text{DEXT} * \text{LENGTH} * \text{NTUBES}$.

COND	Input	Tube Thermal conductivity

If $\text{COND} \leq 0$, then the tubes thermal resistance is ignored.

GEOMFACT	Input	Geometric factor for heat transfer coefficient

The heat exchange is controlled by radiation:

$$\text{LOAD} = \text{CFH} \cdot \text{GEOMFACT} \cdot \sigma \cdot \text{AREA} \cdot (\text{TFUMES}^4 - \text{TWALL}^4)$$

Where : σ is the Boltzmann constant ($5.667 \cdot 10^{-8} \text{ W/m}^2 \text{ K}^4$)

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

STEAMP	Input	Drum pressure (can be manipulated)

If $\text{STEAMP} \leq 0$, the feed water pressure is used.

CIRCFLOW	Input	Flow rate in the vaporiser (can be manipulated)

If $\text{CIRCFLOW} \leq 0$, or if it is too low, the flow rate in the vaporiser loop is calculated in order to match the specified outlet vapour fraction (ALF).

ALF	Input	Outlet vapour fraction (can be manipulated)

This parameter is used only if CIRCFLOW is too low or ≤ 0 .

If $\text{ALF} \leq 0$, the default value 0.5 is used.

SETPOINT	Input	Air excess or fumes O2/CO2 contents (can be manipulated)

The SETPOINT is used for controlling the air flow rate, according to the air flow calculation mode defined by the MODEXC parameter.

PURGE **Input** **Purge as a fraction of feed water (can be manipulated)**

Fraction of the feed water flow rate which is extracted through the purge outlet stream.

CFH **Input** **Heat transfer correction factor (can be manipulated)**

This parameter is a correction factor of the heat transfer coefficients inside the tubes.

EPS **Input** **Relative precision for thermal balance**

If $EPS \leq 0$, the default value 10^{-4} is used.

SECTION 10: HTX070 : SHELL AND TUBES HEAT EXCHANGER

PURPOSE

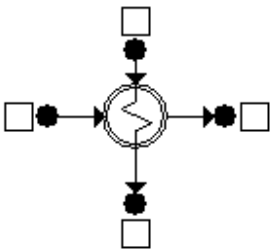
The **HTX070** is a rigorous simulation model of a shell and tubes heat-exchanger device used in vaporiser, condenser or traditional heat exchange mode. The exchanger is basically a counter-current exchanger with possibly several tube passes and/or baffles. The heated fluid can either go through the tubes or through the shell.

When a shift phase does occur, the exchanger is divided up into 4 sections (a section being defined as a zone without phase change) and is calculated section after section. The heat load in each zone is calculated as:

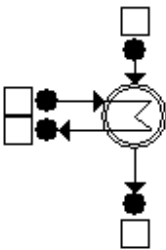
$$Q = U_i.A_i.F.DTLM_i$$

Where :	U_i	Transfer coefficient for the considered section defined by : $1/U_i = FF + 1/(CFH.U_{ci})$
	U_{ci}	Clean heat transfer coefficient as computed from correlations (kW/(K.m ²))
	FF	Fouling factor to be identified.
	CFH	Correction coefficient to be identified.
	A_i	Heat Transfer area for the considered section (computed).
	F	Factor of deviation from counter-flow rate, according to the number of passes at the tubes and shell.
	DTLM _i	Log Mean Temperature Difference for the considered section.

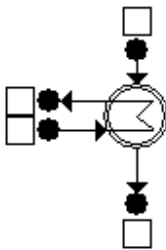
SYMBOLS AND CONNECTION RULES



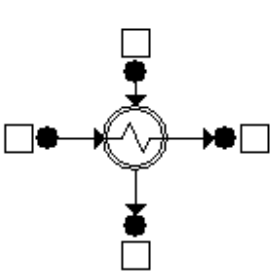
Up/Down



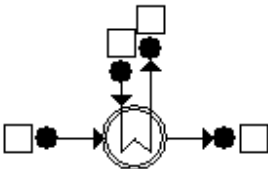
Left/Left Co-Current



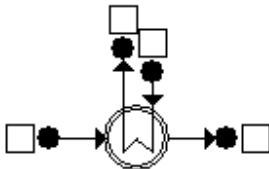
Left/Left Counter-Current



Left/Right



Up/Up Co-Current



Up/Up Counter-Current

Inlet STREAMS	1	Material stream (any state) circulating through the shell
	2	Material stream (any state) circulating through the tubes
Outlet STREAMS	1	Material stream (Liquid/Vapour) exiting from the shell
	2	Material stream (Liquid/Vapour) exiting from the tubes

The corresponding inlet and outlet streams must use the same Thermod.

UNIT PARAMETERS (SUMMARY)

TUBLAYOU	Input	Tube layout
DISPHV	Input	Position of device
BAFLAYOU	Input	Longitudinal baffle layout
KPRINT	Input	Print-out Level
NTUBES	Input	Number of tubes
NTUBPASS	Input	Number of tubes passes
LENGTH	Input	Tubes length
TD	Input	Tubes outside diameter
TDIN	Input	Tubes inside diameter
COND	Input	Tubes thermal conductivity
TUBPITCH	Input	Tube pitch
SHELLDIA	Input	Shell diameter
BAFPITCH	Input	Baffle pitch
CFH	Input	transfer correction factor
FF	Measurable	Fouling factor
CFDPSHL	Measurable	Shell pressure drop correction factor
CFDPTUB	Measurable	Tubes pressure drop correction factor
EPSA	Input	Relative precision on used area
USEDAREA	Input	Fraction of area actually used

UNIT PARAMETERS

TUBLAYOU	Input	Tube layout
TUBLAYOU = 1 -		Pitch type = square
TUBLAYOU = 2 -		Pitch type = triangular
TUBLAYOU = 3 -		Pitch type = square rotated
TUBLAYOU = 4 -		Pitch type = triangle with cleaning lanes
DISPHV	Input	Position of device
DISPHV = 0	horizontal	
DISPHV = 1	vertical.	

BAFLAYOU **Input** **Longitudinal baffle layout**

BAFLAYOU = 0 No horizontal baffle.

BAFLAYOU = 11 One horizontal baffle with vertically cut segmental baffles.

BAFLAYOU = 12 One horizontal baffle with horizontally cut segmental baffles.

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

NTUBES **Input** **Number of tubes**

NTUBPASS **Input** **Number of tubes passes**

If BAFFLE LAYOUT = 0 NTUBPASS must be even or equal to 1.

If BAFFLE LAYOUT > 0 NTUBPASS must be even.

LENGTH **Input** **Tubes length**

TD **Input** **Tubes outside diameter**

TDIN **Input** **Tubes inside diameter**

COND **Input** **Tubes thermal conductivity**

If COND <= 0, then the tubes resistance is ignored.

TUBPITCH **Input** **Tube pitch**

SHELLDIA **Input** **Shell diameter**

BAFPITCH **Input** **Baffle pitch**

CFH **Input** **transfer correction factor**

If CFH < 0, then 1 is chosen as the default value.

$$1/U_i = FF + 1/(CFH \cdot U_{ci})$$

Where : U_i heat transfer coefficient for the considered section.

U_{ci} heat transfer coefficient for clean exchanger computed from correlations (kW/(K.m²)).

FF **Measurable** **Fouling factor**

The fouling factor enables you to account for the presence of adhering film or scale onto the surfaces.

This value may be negative.

CFDPSHL **Measurable** **Shell pressure drop correction factor**

If $CFDPSHL < 0$, then no correction is applied.

CFDPTUB **Measurable** **Tubes pressure drop correction factor**

If $CFDPTUB < 0$, then no correction is applied.

EPSA **Input** **Relative precision on used area**

If $EPSA \leq 0$, the default value 10^{-3} is used.

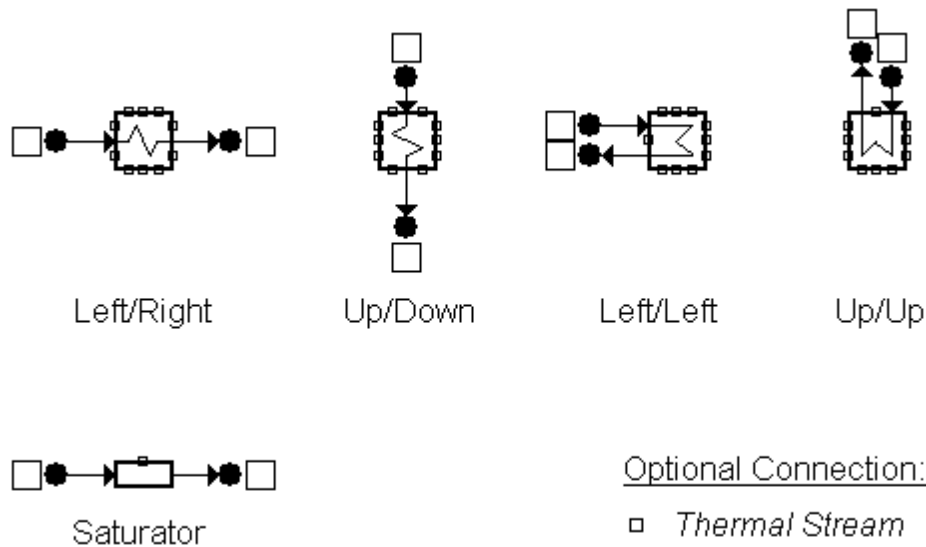
USEDAREA **Input** **Fraction of area actually used**

SECTION 11: HTX100 : HEAT EXCHANGER (ONE SIDE)

PURPOSE

Calculates the energy balance on one side of a heat exchanger where a stream is heated up or cooled down. Different calculation modes are available. The temperature and composition of the inlet stream is always known.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS 1 material stream of any state

Outlet STREAMS 1 material stream of any state, but using the same Thermod as the inlet stream.

In addition one thermal stream can be connected to the unit either as inlet or outlet stream. If the stream is connected as inlet (or outlet) stream, but heat is produced (or used), the thermal stream load will be negative. If no thermal stream is connected, then the heat load will be zero.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation mode
KPRINT	Input	Print-out Level
TOUT	Measurable	Outlet temperature
ALFOUT	Measurable	Outlet vapour fraction
DP	Measurable	Reference pressure drop
DPMASS	Input	Reference mass flow rate
DPEXP	Input	Exponent for flow rate dependence
DPRO	Input	Reference density for side

UNIT PARAMETERS

MODE	Input	Calculation mode
MODE=1	Specified: Flow,Tout	Calculated: Load
MODE=2	Specified: Flow,Tout,Alfout	Calculated: Load
MODE=3	Specified: Flow,Load	Calculated: Tout
MODE=4	Specified: Load,Tout	Calculated: Flow
MODE=5	Specified: Load,Tout,Alfout	Calculated: Flow

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

TOUT	Measurable	Outlet temperature
ALFOUT	Measurable	Outlet vapour fraction

Depending on the MODE, these parameters will be used to fix the state of the outlet stream.

DP	Measurable	Reference pressure drop
DPMASS	Input	Reference mass flow rate
DPEXP	Input	Exponent for flow rate dependence
DPRO	Input	Reference density for side

If $DP < 0$	the outlet stream pressure is set to $ DP $
If $DP = 0$	the outlet stream pressure is equal to the inlet pressure
If $DP > 0$	the outlet stream pressure is calculated as:

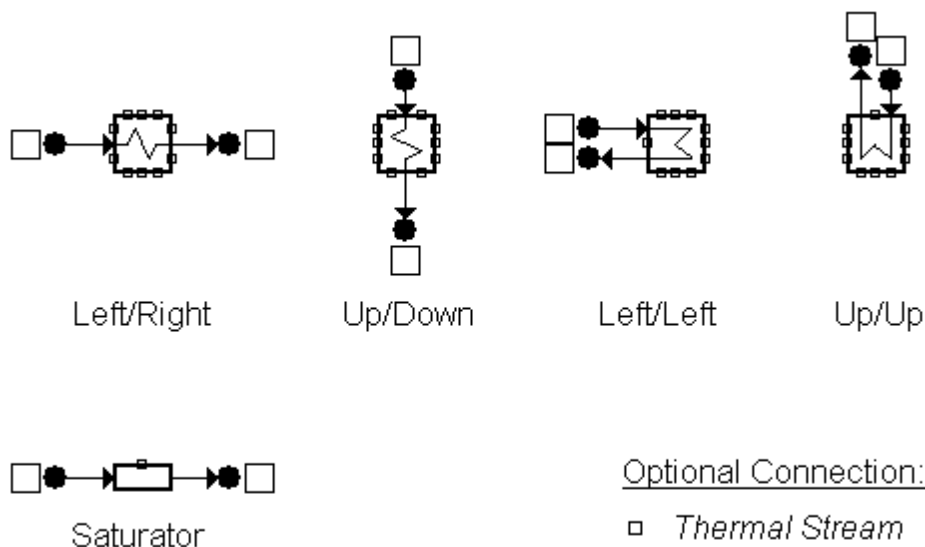
$$POUT = PIN - DP (MASSF/DPMASS)DPEXP (DPRO/RO)$$

SECTION 12: HTX110 : STREAM STATE TOOLBOX

PURPOSE

The HTX100 allows modifying the state of the outlet stream, either by setting directly a combination of its temperature, pressure or vapour fraction or by specifying a state variation (pressure drop or enthalpy change) between the inlet and the outlet stream.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS 1 material stream of any state

Outlet STREAMS 1 material stream of any state, but using the same Thermod as the inlet stream.

In addition one thermal or mechanical stream can be connected to the unit either as inlet or outlet stream. If the stream is connected as inlet (or outlet) stream, but heat or energy is produced (or used), the thermal or mechanical stream load or power will be negative.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation mode
T	Measurable	Outlet temperature
P	Measurable	Outlet pressure
DT	Measurable	Temperature difference
DP	Measurable	Pressure difference
DH	Measurable	Enthalpy difference
ALFOUT	Measurable	Outlet vapour fraction

UNIT PARAMETERS

MODE	Input	Calculation mode
MODE=1	Specified: DT,DP	Calculated: DH
MODE=2	Specified: DP,DH	Calculated: DT
MODE=3	Specified: T,P	Calculated: DH
MODE=4	Specified: P,DH	Calculated: DT
MODE=5	Specified: P,ALFOUT	Calculated: DT,DH
MODE=6	Specified: T,ALFOUT	Calculated: DP,DH
MODE=7	Specified: DP,T	Calculated: DH
MODE=8	Specified: DP,ALFOUT	Calculated: DT,DH
MODE=9	Specified: DT,P	Calculated: DH
MODE=10	Specified: DT,ALFOUT	Calculated: DP,DH

T	Measurable	Outlet temperature
P	Measurable	Outlet pressure
DT	Measurable	Temperature difference
DP	Measurable	Pressure difference
DH	Measurable	Enthalpy difference
ALFOUT	Measurable	Outlet vapour fraction

When MODE=1,4,7,9, the parameter ALFOUT is also used for single compound stream, as the temperature and pressure do not explicitly define its state in the liquid/vapour region.

SECTION 13: INCREM : LOOP CONTROLLER

PURPOSE

The **INCREM** unit enables to calculate several times in a row a given PFD, while incrementing some variables (maximum 10) at each iteration. This is particularly useful to make a parametric sensitivity analysis on a model. If more than 10 variables must be modified or for non standard incrementation strategies, you can use a FLEX program associated to the **INCREM** unit (or any other unit).

As other convergence units, the **INCREM** unit must be the last unit of the PFD.

The unit **INCREM** is usually combined with a **REPORT** unit that prints some key values in a separate file (1 line per iteration) for subsequent printing or plotting.

If no unit of type **REPORT** is used, you can follow the results at each iteration by browsing the trace file (r3s). The amount of information printed in this trace file varies with the value of the **KPRINT** parameter of the different units included in the PFD.

SYMBOLS AND CONNECTION RULES



You cannot connect any stream to an **INCREM** unit.

UNIT PARAMETERS (SUMMARY)

TYPE	Input	Type of incrementation
KPRINT	Input	Print-out Level
ITMAX	Input	Number of iterations
NVAR	Input	Number of variables
FLASHi	Input	Stream Initialisation Mode (i=1,10)

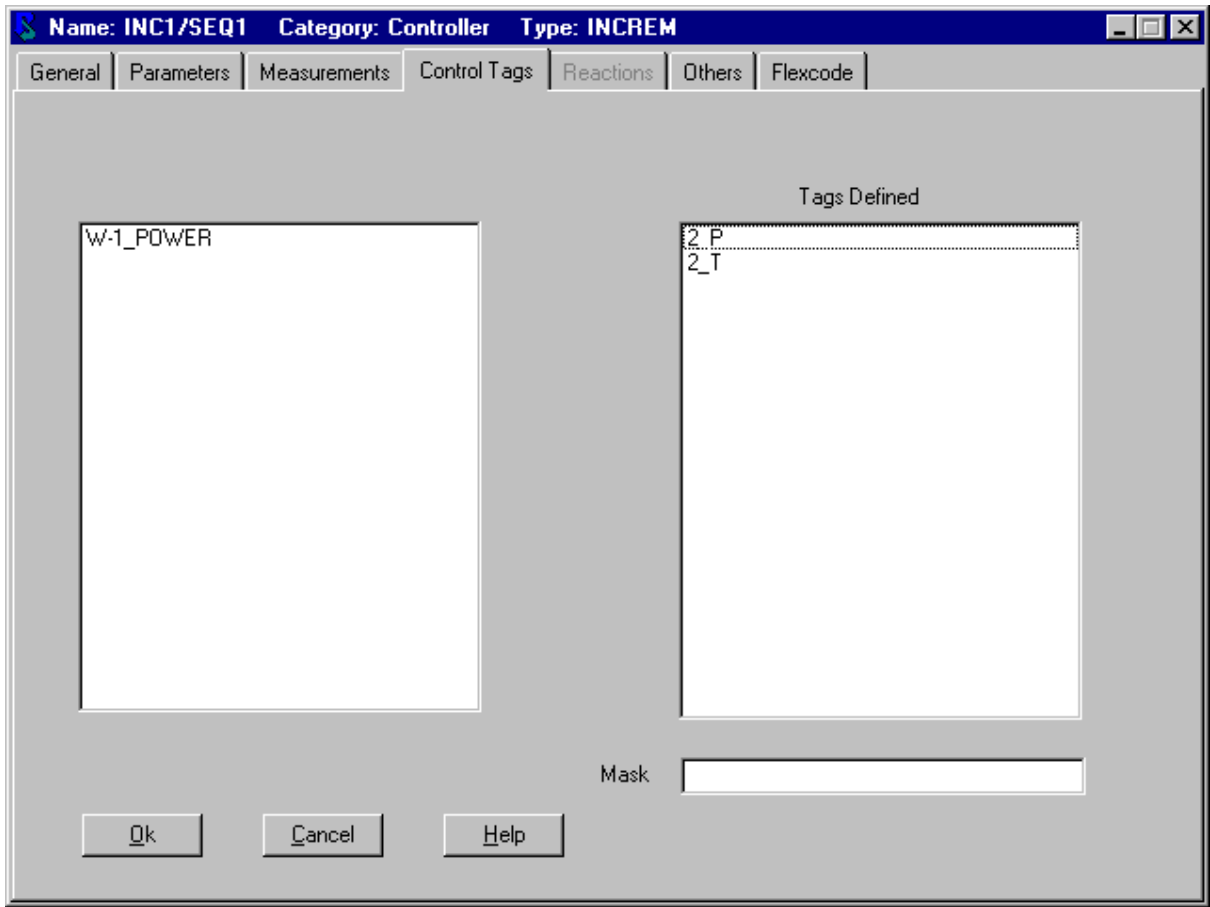
SPECIFYING THE VARIABLES TO BE INCREMENTED

The variables to be incremented are defined via the **Control Tags** page of the Unit Notebook. The list of existing Tags is shown in the **Tags Defined** list. You can use the **Mask** field to limit the number of tags displayed in this list. The following types of Tags are supported:

Material Streams:	T, P, VAPF, MIXF, H, MOLF, MASSF, Partial Flowrates
Thermal Streams:	LOAD
Mechanical Streams	POWER
Units	Any measurable input parameter

The measured value of the Tags will be used for incrementing the corresponding variables at each iteration.

Note: Some of the unit parameters can be either input (given) or output (result) depending on their **MODE** of operation



UNIT PARAMETERS

TYPE	Input	Type of incrementation
1 : + (addition)		at each iteration the selected variables will be increased by a given value
2 : * (multiplication)		at each iteration the selected variables will be multiplied by a given value

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

ITMAX **Input** **Number of iterations**

The parameter ITMAX determines how many times the PFD will be executed.

NVAR **Input** **Number of variables**

The parameter NVAR determines the number of variables that are incremented at each iteration.

FLASHi **Input** **Stream Initialisation Mode (i=1,10)**

When the manipulated variable belongs to a material stream, the parameter FLASHi indicates how to reinitialise the stream:

- 1 Stream state is not calculated
- 0 Material balance only; the stream enthalpy is set to -1
- 1 Stream state is based on its pressure and temperature (EPT)
- 2 Stream state is based on its pressure and vapour fraction (EPA)
- 3 Stream state is based on its temperature and vapour fraction (ETA)
- 4 Stream state is based on its enthalpy and pressure (EPH)

SECTION 14: MAT010 : CONVERGENCE PROMOTER

PURPOSE

Promotes the convergence of the torn variables of STREAMS using the WEGSTEIN method.

Several streams can be handled simultaneously.

Convergence promotion on molar enthalpy is preferred to promotion on temperature, so as to avoid problems in case of a phase change in a single component mixture (pure compound).

Estimated values for the torn variables are provided in the outlet STREAMS according to the situation. The process is then calculated to generate process answers which are available in the inlet STREAMS.

The MAT010 UNIT compares the starting values and the results and checks that convergence has been obtained. If convergence is not obtained, new values for the unknowns are proposed and the process calculation is then resumed. This process is repeated until convergence is achieved.

The MAT010 UNIT must be the last UNIT in a simulation sequence (SEQUEN UNIT).

The number of torn variables is limited to 20.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	Maximum number : 10.
	Type MAT.
	Any state may be used.

Outlet STREAMS	Maximum number : 10.
	Type MAT.
	Any state may be used.

Associated THERMODs :	Must be identical for inlet and outlet i.

UNIT PARAMETERS (SUMMARY)

TEARTY <i>i</i>	Input	Tear type for inlet stream <i>i</i>
KPRINT	Input	Print-out Level
ITER	Result	Iteration in progress
ITMAX	Input	Maximum number of iterations
ITCON	Result	Convergence indicator
FREQ	Input	Frequency of the WEGSTEIN algorithm
RELAX	Input	Relaxation factor
ANG1	Input	Angular coefficient
ANG2	Input	Angular coefficient
EPSH	Input	Absolute precision for "molar enthalpy" tears
EPSZ	Input	Relative precision for "flowrate" or "composition" tears

UNIT PARAMETERS

TEARTY*i* **Input** **Tear type for inlet stream *i***

- 1 Only the total molar flowrate is torn.
- 2 Only the molar enthalpy is torn.
- 3 The total molar flowrate and molar enthalpy are torn.
- 4 The molar compositions are torn.
- 5 The partial molar flow rates and the molar enthalpy are torn.
- 6 The molar compositions and the molar enthalpy are torn.
- 7 The partial molar flow rates are torn.

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

ITER **Result** **Iteration in progress**

Iteration in progress (initial value has no influence on simulation).

ITMAX **Input** **Maximum number of iterations**

Maximum number of iterations.

If ITMAX = 0, then ITMAX is set to 30.

If $ITMAX > 0$, then $ITMAX$ is the maximum number of iterations :

The simulation is halted if convergence is not achieved within the $ITMAX$ number of iterations.

If $ITMAX$ sign is negative :

The simulation proceeds either at convergence, or after the $ITMAX$ number of iterations has been reached.

ITCON	Result	Convergence indicator

Convergence indicator.

$ITCON = 1$ if convergence is achieved.

$ITCON = 0$ while convergence is not achieved.

The initial value has no influence on the calculation

FREQ	Input	Frequency of the WEGSTEIN algorithm

Application frequency of the WEGSTEIN algorithm. Successive (relaxed) substitution is applied for other iterations.

Advice : use $FREQ = 1$

RELAX	Input	Relaxation factor

Relaxation factor :

$$XCOR = X * (1 - RELAX) + F(X) * RELAX.$$

Advice : use $RELAX = 1$

ANG1	Input	Angular coefficient

Angular coefficient.

If the angular coefficient of the WEGSTEIN segment lies between $ANG1$ and 1, it is set to $ANG1$ in order to avoid abnormal extrapolation.

Advice : use $ANG1 = 0.9$

ANG2	Input	Angular coefficient

Angular coefficient.

If the angular coefficient of WEGSTEIN segment lies between 1 and $ANG2$, it is set to $ANG2$ in order to avoid abnormal extrapolation.

Advice : use ANG2 = 1.1

EPSH	Input	Absolute precision for "molar enthalpy" tears

Absolute precision for "molar enthalpy" tears.

If $ABS(F(Hmol) - Hmol) \leq EPSH$ (kJ/kmol) convergence is achieved.

Advice : use EPSH = 0.5

EPSZ	Input	Relative precision for "flowrate" or "composition" tears

Relative precision for "flowrate" or "composition" tears.

If $ABS(F(X) - X) \leq EPSZ * X$. convergence is achieved.

If $EPSZ * X$ is smaller than 1.D-10, $EPSZ * X$ is replaced by 1.D-10.

Advice : use EPSZ = 1.d-04

SECTION 15: MAT020 : CONVERGENCE PROMOTER

PURPOSE

This UNIT performs simultaneous convergence promotion, on a maximum of 10 STREAMS.

According to the value of the MODE parameter, tears are :

- Total (they concern molar enthalpy, pressure and partial molar flow rates),

OR

- Material (only molar flow rates are concerned).

The mathematical algorithm is based on the BROYDEN method, which takes into account the interactions in the systems of non-linear equation systems.

As this UNIT is a convergence promotion UNIT, it must be the last in a group of UNITs gathered into a SEQUEN type UNIT.

Prior to starting the simulation, outlet STREAMS and MIXDATs must provide a feasible approximation of the solution.

WARNING

The NDIM's (number of torn variables) first iterations are used to calculate the Jacobian matrix by altering the variables one after the other.

SYMBOLS AND CONNECTION RULES

Inlet STREAMS	Maximum number : 10. Type MAT.
Outlet STREAMS	Maximum number : 10. Type MAT.
Associated THERMODs :	Must be identical for inlet and outlet i.

UNIT PARAMETERS

MODE	Input	Computation Mode

MODE = 0	-	If tears only concern flow rates, enthalpy is set to -1 whilst T and P remain unchanged.
MODE = 1	-	If tears are total (enthalpy, pressure and flow rates). An isenthalpic calculation is performed with the new values of the variables.

DMIN,DMAX,TMIN,TMAX,PMIN,PMAX Input LOWER,UPPER BOUNDS

Partial molar flow rates, temperatures and pressures, lower and upper bounds.

If T minimum ≤ 0 , the default value of 100 K is taken by the program.

If T maximum ≤ 0 , the default value of T Min + 500 K is taken by the program.

If P minimum ≤ 0 , the default value of 1.D-02 bar is taken by the program.

If P maximum ≤ 0 , the default value of P Min + 100 bar is taken by the program.

KPRINT Input Print-out Level

Controls the amount of information printed in the trace file.

ITER Input Iteration in progress

Iteration in progress (initial value has no influence on simulation).

ITMAX Input Maximum number of iterations

Maximum number of iterations.

If ITMAX = 0, then ITMAX is set to 30.

If ITMAX > 0, then ITMAX is the maximum number of iterations :

The simulation is halted if convergence is not achieved within the ITMAX number of iterations.

If ITMAX sign is negative :

The simulation proceeds either at convergence, or after the ITMAX number of iterations has been reached.

ITCON Result Convergence indicator

Convergence indicator.

ITCON = 1 if convergence is achieved.

ITCON = 0 while convergence is not achieved.

The initial value has no influence on the calculation

EPS Input Precision required for convergence criteria

Precision required for convergence criteria.

Sum of square residue $\leq$ EPS

If EPS $\leq$ 0 default value $1.10 \cdot 10^{-6}$ is taken by the program.

DELTA Input Value used at first call to calculate the research direction

Value used at first call to calculate the research direction :

Newton if it is high,

steepest descent if low value.

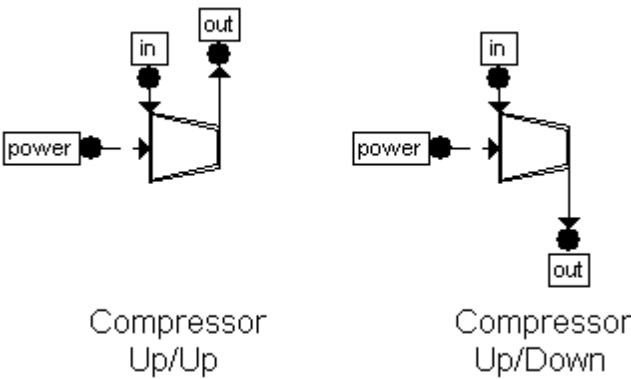
If DELTA $\leq$ 0, the default value of 10 is taken by the program.

SECTION 16: MEC010 : GAS COMPRESSOR

PURPOSE

The MEC010 unit is a simulation model of a gas compressor, based on a given isentropic efficiency. Various calculation modes are available (specified outlet pressure, compression ratio or power). The stream flow rate can also be adjusted to match a given outlet pressure and compressor power.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Material stream (any state)
	2	Mechanical stream
Outlet STREAM	1	Material stream (Liquid/vapour state) using the same Thermod as the material inlet stream

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation Mode
KPRINT	Input	Print-out Level
ISENTEFF	Measurable	Isentropic efficiency
PRATIO	Measurable	Compression ratio
POUT	Measurable	Outlet pressure

UNIT PARAMETERS

MODE	Input	Calculation Mode

The inlet stream composition, pressure and temperature are always known, as well the compressor isentropic efficiency.

MODE=1	Specified: Flow,PRATIO	Calculated: Tout,Power
MODE=2	Specified: Flow,POUT	Calculated: Tout,Power
MODE=3	Specified: Flow,Power	Calculated: Tout,Pout
MODE=4	Specified: Power,POUT	Calculated: Tout,Flow

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

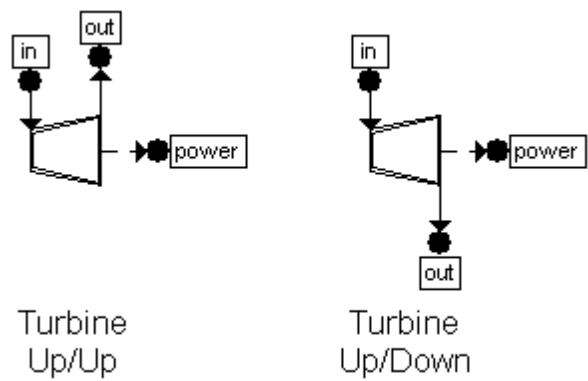
ISENTEFF	Measurable	Isentropic efficiency
PRATIO	Measurable	Compression ratio
POUT	Measurable	Outlet pressure

SECTION 17: MEC030 : TURBINE

PURPOSE

The MEC030 unit is a simulation model of a turbine, based on a given isentropic efficiency. Various calculation modes are available (specified outlet pressure, expansion ratio or power). The stream flow rate can also be adjusted to match a given outlet pressure and expansion power.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Material stream (any state)
Outlet STREAM	1	Material stream (Liquid/vapour state) using the same Thermod as the material inlet stream
	2	Mechanical stream

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation Mode
KPRINT	Input	Print-out Level
ISENTEFF	Measurable	Isentropic efficiency
PRATIO	Measurable	Expansion ratio
POUT	Measurable	Outlet pressure

UNIT PARAMETERS

MODE	Input	Calculation Mode

The inlet stream composition, pressure and temperature are always known, as well the compressor isentropic efficiency.

MODE=1	Specified: Flow,PRATIO	Calculated: Tout,Power
MODE=2	Specified: Flow,POUT	Calculated: Tout,Power
MODE=3	Specified: Flow,Power	Calculated: Tout,Pout
MODE=4	Specified: Power,POUT	Calculated: Tout,Flow

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

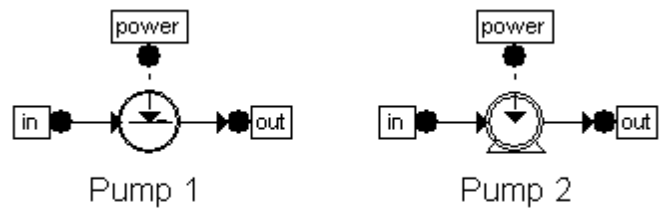
ISENTEFF	Measurable	Isentropic efficiency
PRATIO	Measurable	Compression ratio
POUT	Measurable	Outlet pressure

SECTION 18: MEC050 : LIQUID PUMP

PURPOSE

The MEC050 unit is a simulation model of a pump, taking into account its volumetric efficiency. The pump can be calculated for a given outlet pressure or a given pressure increase.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Material stream (any state)
	2	Mechanical stream
Outlet STREAM	1	Material stream (any state involving a Liquid phase) using the same Thermod as the material inlet stream

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation Mode
POUT	Measurable	Outlet pressure or pressure variation
EFFI	Measurable	Pump efficiency
KPRINT	Input	Print-out Level

UNIT PARAMETERS

MODE	Input	Calculation Mode
MODE = 1		Outlet pressure is given.
		Outlet temperature and pump power are calculated.
MODE = 2		Pressure increase is given.
		The outlet temperature and the pump power are calculated.

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

POUT **Measurable** **Outlet pressure or pressure variation**

The signification of this parameter depends on the value of MODE.

EFFI **Measurable** **Pump efficiency**

SECTION 19: MEC110 : GAS COMPRESSOR

PURPOSE

UNIT MEC110 simulates a gas compressor with a constant rotation speed.

The calculation depends upon :

1. The characteristic curve for isentropic efficiency, as a function of the flow rate factor.
2. The characteristic curve for isentropic head, as a function of the flow rate factor.

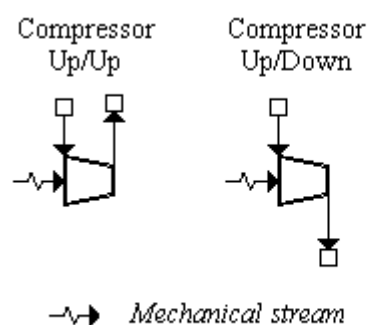
The flow rate factor is an undimensional parameter function of the inlet volumetric flow rate.

These curves are computed from the available information :

- compression ratio and
- power supplied to the compressor shaft.

Compressor control is done by a throttle valve controlled by the outlet pressure.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS

Number : 2.
Type (1) MAT, and (2) MECSTRM.
State : V or LV.

Outlet STREAM

Number : 1.
Type MAT.
State: V or LV.

Associated THERMOD

First Inlet and outlet STREAMS must have the same THERMOD.

UNIT PARAMETERS (SUMMARY)

KPRINT	Input	Print-out Level
RISEN	Result	COMPUTED ISENTROPIC EFFICIENCY
ETAORG	Input	ORGANIC EFFICIENCY
VITNOM		NOMINAL OPERATING SPEED
VITMAX		MAXIMUM ALLOWED SPEED
PHINOM		NOMINAL FLOWRATE COEFFICIENT
DIAM		WHEEL DIAMETER
EFFIC		MECHANIC EFFICIENCY
PHI	Result	COMPUTED VALUE FOR FLOWRATE COEFFICIENT
PSIC	Result	COMPUTED VALUE FOR HEAD COEFFICIENT
ALARM		WARNING FOR COMPRESSOR STATE
RIS1,RIS2,RIS3,RIS4	Input	ISENTROPIC EFFICIENCY PARAMETER
PSI1,PSI2,PSI3,PSI4	Input	Parameters to define the head factor curve
EPS	Input	Precision required at convergence
PHIMIN.1 & PSIMIN.1		
PHIMIN.2 & PSIMIN.2		

UNIT PARAMETERS

KPRINT Input Print-out Level

Controls the amount of information printed in the trace file.

RISEN Result COMPUTED ISENTROPIC EFFICIENCY

Isentropic efficiency, for more information see Miscellaneous Parameters Table.

ETAORG Input ORGANIC EFFICIENCY

The Organic Efficiency is the correction factor for the Isentropic Efficiency used to adjust the calculated outlet temperature.

VITNOM NOMINAL OPERATING SPEED

Nominal operating speed measured in rpm.

VITMAX MAXIMUM ALLOWED SPEED

Maximum allowed speed measured in rpm.

PHINOM **NOMINAL FLOWRATE COEFFICIENT**

This coefficient is not necessary in every case.

DIAM **WHEEL DIAMETER**

Wheel diameter in meters (not necessary in every case).

EFFIC **MECHANIC EFFICIENCY**

The mechanical efficiency is a correction on the head factor.

PHI **Result** **COMPUTED VALUE FOR FLOWRATE COEFFICIENT**

Computed value for flow rate coefficient is measured in m³/hr.

PSIC **Result** **COMPUTED VALUE FOR HEAD COEFFICIENT**

Computed value for isentropic head coefficient.

$$PSIC = PSI * EFFIC$$

ALARM **WARNING FOR COMPRESSOR STATE**

Warning parameter for compressor state.

In compressors, there is a minimum capacity at every speed, below which operation is unstable. This instability (called pumping or surge) is to be avoided if possible:

- if surging mode then Warning is set to 1,
- if operation is safe Warning is set to 0.

Parameters to define the isentropic efficiency curve.

RIS1,RIS2,RIS3,RIS4 **Input** **ISENTROPIC EFFICIENCY PARAMETER**

$$RISEN = \frac{(RIS1 + RIS2 * PHI + RIS3 * PHI^2)}{(PHI - RIS4)}$$

Where : PHI = Flowrate Coefficient.

PSI1,PSI2,PSI3,PSI4 **Input** **Parameters to define the head factor curve**

$$HEADCOEFFICIENT = \frac{(PSI1 + PSI2 * PHI + PSI3 * PHI^2)}{(PHI - PSI4)}$$

$$HEADCOEFFICIENT = H_{is} * \frac{g}{U^2}$$

Where : PHI = Flow rate Coefficient.
 U = Radial velocity of wheel (m/s).

$$U = \frac{\pi D N}{60}$$

N : rpm D : m

EPS Input Precision required at convergence

Precision required at convergence :

The default value is 1. 10-05

relative precision on pressure if EPS > 0,

absolute precision on pressure if EPS < 0.

PHIMIN.1 & PSIMIN.1

Define the point 1 of the compressor surge line.

PHIMIN.2 & PSIMIN.2

Define the point 2 of the compressor surge line.

SECTION 20: MEC130 : GAS COMPRESSOR

PURPOSE

Simulates a gas compressor on the basis of the following knowledge :

- its characteristic curves e.g. flowrate-mechanical power, flowrate-compression ratio or outlet pressure,
- its outlet conditions at the nominal point i.e. the point where the compressor works at its maximum efficiency that is with the minimum losses.

These conditions are :

- The pressure,
- The temperature,
- The weight flow rate,
- The required power and
- The rotation speed.

The characteristic curves are converted to curves defined by the variables of the polytropic efficiency and the pressure factor as a function of the flow rate factor.

$$ETAPI = f(PHI) = A + B.PHI + C.PHI^2 + \_F(D;PHI) + E.lnPHI$$

$$PSY = f(PHI) = A' + B'.PHI + C'.PHI + \_F(D';PHI) + E'.lnPHI$$

Where : ETAPI Is the polytropic efficiency.

 PHI Is the flow rate factor.

 PSY Is the pressure factor.

The flow rate factor PHI is defined from the equation :

$$PHI = Q/N$$

and the pressure factor PSY from the equation :

$$PSY = H_p/N^2$$

Where : N = Rotation speed (rps).

 Q = Volumetric flow rate (m³/s).

 H_p = Polytropic head (kJ/kg).

The polytropic head H_p is determined from the following equation :

$$H_p = P \_F(Q;GKG) \_F(m;m-1) \_B\_bc\_((Rc\_S\_Up5(\_F(m-1;m);) -1)$$

Where :	P	=	Inlet pressure (kPa).
	GKG	=	Mass flowrate (kg/s).
	Rc	=	Pressure ratio (discharge pressure/inlet pressure).
	m	=	Polytropic exponent.

The m value is a function of the polytropic efficiency η_p

$$\frac{P}{P_0} = \left(\frac{GKG}{GKG_0} \right)^{\frac{1}{m-1} \cdot \eta_p}$$

Where :	k	=	The ratio of specific heats (C_p/C_v).

Furthermore, the polytropic efficiency must satisfy the gas-power requirement equation given by :

$$PWR = \frac{P}{\eta_p} \cdot \left(\frac{GKG}{GKG_0} \right)^{\frac{1}{m-1} \cdot \eta_p}$$

Where :	PWR	=	Required power (kW).
	η_{mec}	=	Mechanical losses factor.

So for each point of the power - discharge pressure characteristic curves, a polytropic head and polytropic efficiency must be calculated that will result in the discharge pressure and power requirements shown on the curves.

TOPOLOGY

Inlet STREAMS	Number : 2.
	Type (1) MAT, and (2) MECSTRM.

Outlet STREAMS	Number : 1.
	Type MAT (LV or V).

Associated THERMODs : Must be identical.

UNIT PARAMETERS

POLYTROPIC EFFICIENCY (ETAPIC)

This polytropic efficiency field cannot be accessed since its value is being computed from the given PHI and the coefficient values on the simulation mode. Only the physical unit field can be accessed for a percentage or an absolute value to be chosen.

NOMINAL MECHANICAL EFFICIENCY (KMECNOM)

Not a necessary field. It already exists in nominal point conditions.

COMPUTATION MODE (MODE)

There are two computation modes available :

MODE = 1 Simulation Mode.

The ETAPI and PSY coefficients are given in the MEC130 UNIT as well as the mechanical losses (if any) and the rotation speed on the nominal point. The outlet pressure and temperature, the required power and polytropic efficiency are calculated. The rotation speed of the actual conditions should also be given in the mechanical stream of the compressor.

Attention: If there are large variations in the inlet conditions of the compressor, the accuracy of the developed efficiency curve may deteriorate substantially due to the volume ratio effects (for multiple compression).

In order to check the validity of the curve for the new conditions a variable Q (TETA) is calculated where :

$$Q = \frac{B_{bc}((F(\mu; k))_{S_{up}})^{0.5}}{B_{bc}((F(\mu; k))_{S_{ref}})^{0.5}}$$

for referenced and new inlet conditions, μ being the molecular weight of the gas stream (kg/kmol). If the value of Q for the new conditions is within 5 % of its value for the old or reference conditions the efficiency curve developed by this procedure is likely to be valid for the new conditions. So, the efficiency curve has the advantage over the characteristic curves of the compressor, in that it can be used for gas streams with different molecular weight.

MODE = 2 Identification Mode.

For the coefficients of the functions (1) and (2).

The coefficients are identified by a PID010 UNIT containing ONLY the MEC130 UNIT.

The EXPDAT object associated to the PID010 UNIT contains :

PHI as an independent (I) variable,

and PSY, ETAPI as measured (M) ones.

The outlet pressure and the compressor's power in the MECSTRM should be given.

CHARACTERISTIC CURVES COEFFICIENTS TABLE :

Col. 1(blank)	Variable for which the coefficients are to be entered. It contains two variables : ■ the polytropic efficiency ETAPI and ■ the pressure factor PSY.

A - E	The values of the corresponding coefficients are to be entered. This table is only taken into consideration for the simulation mode (MODE = 1).

ETAPI ADJUSTMENT

This value represents an adjustment coefficient for the calculated polytropic efficiency.

PSY ADJUSTMENT

This value represents an adjustment coefficient for the calculated variable PSY.

LIMIT CONDITIONS

Limit working conditions are needed in order to define the limit line and surge line of the characteristic curves.

These are defined from the following variables :

MINIMAL AND MAXIMAL RPM (RPMMIN & RPMMAX)

Rotation speed limits of the compressor.

SURGE LINE and LIMIT LINE

Weight flowrate (physical units can be chosen) and pressure ratio.

Two sets of values are needed to define each line. They are optional, but if given a verification of the compressor being within function boundaries is permitted in simulation mode.

CONDITIONS AT NOMINAL POINT

Working conditions taken into consideration are the following :

OUTLET TEMPERATURE	(TOUTNOM)
OUTLET PRESSURE	(POUTNOM)
WEIGHT FLOWRATE	(MASSFNOM)
POWER	(PWRNOM)
RPM	(RPMNOM)
MECHANICAL LOSSES	(LOS NOM)
MECHANICAL EFFICIENCY	(KMECNOM)
TETA (Q)	(TETA)

These also depend on the MODE :

MODE = 1 Simulation Mode.

These values are not necessary except for the mechanical losses and the rotation speed at the nominal point that are needed for the calculation of the mechanical losses at the given conditions. Also, if a value for TETA is given, the program will perform a test on the validity of the curves for the inlet conditions.

MODE = 2 The first 5 values i.e. the highlighted ones are needed and the rest calculated.

CALCULATION INDICATIONS

PRECISION FOR PRESSURE (EPS)

Defines the precision for outlet pressure estimation.

PRINT-OUT LEVEL (KPRINT)

Defines the Print-out options.

SECTION 21: MIS010 : UTILITY MODEL

PURPOSE

This unit is obsolete. Use unit HTX110 instead.

SECTION 22: MIX010 : MATERIAL STREAMS MIXER

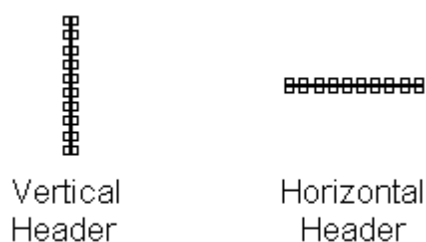
PURPOSE

Calculates the mass and energy balances of a mixer. The outlet stream pressure will be the minimum of the inlet streams pressure.

Different calculation modes are available: straight forward mixing of the inlet streams or control of some inlet flow rates to reach a required flow rate and/or temperature of the outlet stream.

A special mode is also available for simple mass balance.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	From 1 up to 10 Material streams of any state
Outlet STREAMS	1 Material stream of any state
Associated THERMODs	The Thermod of the outlet stream must contain all compounds that actually appear in any of the inlet streams.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation Mode
KPRINT	Input	Print-out Level
ALFOUT	Measurable	Outlet stream vapour fraction (can be controlled by a controller)

UNIT PARAMETERS

MODE	Input	Calculation Mode

The temperature and the composition of every inlet streams are known. Different calculation modes

can be selected:

- 1 Mass and energy balance
- 2 Mass balance only. In this case the state of the outlet stream is left unchanged.
- 3 The flow rate of the second inlet stream is calculated to obtain the temperature specified in the outlet stream data.
- 4 The flow rate of the second inlet stream is calculated to obtain the flow rate specified in the outlet stream data.
- 5 The flow rates of both inlet streams are calculated to obtain the temperature and flow rate specified in the outlet stream data.

When the set-points given in the outlet streams can not be met (Mode =3, 4 and 5), the simulation is stopped.

KPRINT Input Print-out Level

Controls the amount of information printed in the trace file.

ALFOUT Measurable Outlet stream vapour fraction (can be controlled by a controller)

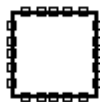
Calculated vapour fraction of the outlet stream.

SECTION 23: MIX020 : MECHANICAL/THERMAL STREAMS MIXER

PURPOSE

Calculates the heat load or the mechanical power of one of the connected STREAMS so that the energy balance is verified.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS From 1 up to 10 Mechanical or Thermal streams

Outlet STREAMS From 1 up to 10 Mechanical or Thermal streams

UNIT PARAMETERS (SUMMARY)

KPRINT	Input	Print-out Level
IPOSIT	Input	Stream to be calculated

UNIT PARAMETERS

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file

IPOSIT	Input	Stream to be calculated

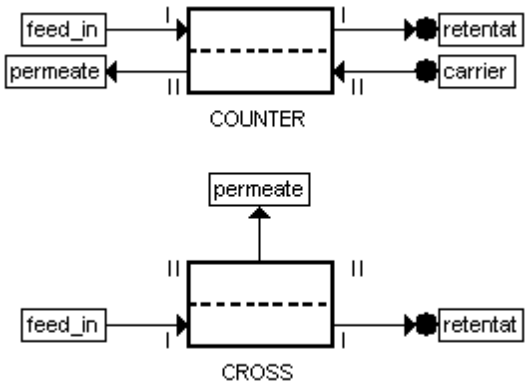
Indicates which stream will be calculated to match the energy balance.

SECTION 24: PERMEA : PERMEATOR

PURPOSE

UNIT PERMEA is a model for a device separating gas across a membrane. Two modes are available: Counter-Current and Cross Plug Flow.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Material stream (V or LV) (I)
	2	Material stream (V or LV) (only in counter-current mode) (II)
Outlet STREAMS	1	Material stream for the retentat (I)
	2	Material stream for the permeate (II)

All the streams must use the same Thermod

UNIT PARAMETERS (SUMMARY)

TYPE	Input	Permeator type
KPRINT	Input	Print-out level
MODE	Input	Calculation mode
NCELL	Input	Number of cells or number of integration steps
SURFACE	Measurable	Total surface area
DPH	Measurable	Pressure drop (HP side)
DPB	Measurable	Pressure drop (BP side) (not used in Cross Plug Flow)
ITMAX	Input	Maximum number of iterations
EPS	Input	Convergence criteria
RDTR	Measurable	Molar ratio for retentate
RDTP	Measurable	Permeation molar ratio
UNIT	Input	Permeances measurement units
PERM1-10	Measurable	Membrane Values

UNIT PARAMETERS

TYPE Input Permeator type

1 : Counter-current

2 : Cross plug flow

KPRINT Input Print-out level

TYPE = 1: 10*n+k

Where : k = impression coefficient (0 to 5) : controls the amount of information printed in the trace file

n = order number of the compound for which an evolution profile is wanted in retentate and permeate streams. The total molar flowrate is also printed.

TYPE = 2: If KPRINT is greater than 0 then the compounds molar flowrate, total molar flowrate and pressure profiles are printed.

MODE Input Calculation mode

Only if ITYPE = 1

0 :	Simulation :	Inlet flowrates and permeances are known. Outlets are calculated.
N :	Regulation :	The "feed_in" stream flowrate is adjusted to obtain an imposed molar fraction for the Nth compound in the retentate. (Retentate composition must be initialised).
-N :	Regulation :	The "feed_in" stream flowrate is adjusted to obtain an imposed molar fraction for the Nth compound in the permeate. (Permeate composition must be initialised).
100 :	Adjustment :	Permeances are calculated to obtain flowrate and compositions fixed in retentate.

NCELL Input Number of cells or number of integration steps

TYPE= 1: Number of cells used in the device model

TYPE= 2: Number of integration steps

(By default = 50, maximum = 100)

SURFACE Measurable Total surface area

Total surface area of the membrane (By default = 100 m²)

DPH Measurable Pressure drop (HP side)

Pressure drop (bar) HP (By hypothesis : linear profile)

DPB Measurable Pressure drop (BP side) (not used in Cross Plug Flow)

Pressure drop (bar) BP (By hypothesis : linear profile)

ITMAX Input Maximum number of iterations

Maximum number of allowed iterations (By default = 50)

EPS Input Convergence criteria

Convergence criteria (By default = 50)

RDTR Measurable Molar ratio for retentate

Retentate molar flowrate/ Sum of inlet molar flowrates

RDTR **Measurable** **Molar ratio for retentate**

Permeate molar flowrate/ Sum of inlet molar flowrates

UNIT **Input** **Permeances measurement units**

-1 : kmol/s/m²/bar

-2 : Nm³/h/m²/bar

> 0 : permeances [kmol/s/m²/bar] = UNIT * following parameters

PERM1-10 **Measurable** **Membrane Values**

Membrane permeance for the different compounds (unit defined by UNIT parameter). These parameters must always be initialised even if MODE = 100.

SECTION 25: OPT070 : OPTIMIZER

PURPOSE

Optimises a user defined objective function subject to equality and non-equality constraints while solving recycle convergence loops at the same time.

As any convergence unit, this Unit must be the last one in the sequence defined by a simulation PFD. Only one unit of type **OPT070** should be active in the simulated PFD and its sub-PFDs.

The objective function will be defined through a flex program pointing to the parameter FOBJ of the unit.

The variables to be optimised are selected by the tags listed in the manipulated tags window. The Safety lower and upper bounds defined on these tags will be taken into account. Up to 100 variables can be handled.

In addition to the recycle constraints defined by the units CUT010 defined in the sub-PFDs, additional constraints can be defined by tags pointing to variables that must be equal to (equality constraints) or greater than (inequality constraints) the measured value of the tags. These tags must be listed in the controlled tag table displayed in the Control Tags page.

SYMBOLS AND CONNECTION RULES



No stream can be connected to this unit.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Precision Mode
KPRINT	Input	Print-out Level
ITMAX	Input	Maximum number of iterations
STEPDB	Input	Initial Step
EPS	Input	Relative tolerance
NEQ	Input	Number of equality constraints
NINEQ	Input	Number of inequality constraints
NVAR	Input	Number of variables
AUTOSCAL	Input	Automatic SCALING
STEPPDX	Input	Step for derivatives
EPSDER	Input	Estimate of Noise
FOBJ	Input	Objective Function
FOBJSCAL	Input	SCALING of objective function

UNIT PARAMETERS

MODE **Input** **Precision Mode**

0 : High precision

1 : Shortcut

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

2 : the unit parameters are printed at each iteration

3 : as 2 + the list of variables and constraints is printed at each major iteration

4 : as 2 + the list of variables and constraints is printed at each iteration

ITMAX **Input** **Maximum number of iterations**

If ITMAX = 0 then ITMAX is set to 30.

If ITMAX > 0 then ITMAX is the maximum number of iterations :

The simulation is stopped if convergence is not achieved within ITMAX iterations.

If ITMAX < 0 The simulation proceeds either at convergence, or after |ITMAX| iterations.

STEPDB Input Initial Step

Default value 0.1

Smaller values will generate smaller initial steps

EPS Input Relative tolerance

Default value 10-7

NEQ Input Number of equality constraints

The first NEQ tags defined in the controlled tags table will be considered as equality constraints. At convergence the variables pointed by these tags will be equal to the value specified on the Tags.

NINEQ Input Number of inequality constraints

The next NINEQ tags defined in the controlled tags table will be considered as inequality constraints. At convergence the variables pointed by these tags will be greater than or equal to the value specified on the Tags.

NVAR Input Number of variables

The first NVAR tags defined in the manipulated tags table will indicate the variables on which the optimiser will act to minimise its objective function while satisfying the specified constraints. The Safety bounds defined on these tags will be taken into account as the maximum range of values for the variables.

The remaining TAGS, if any, will be ignored.

AUTOSCAL Input Automatic SCALING

0 : User defined scaling factors

1 : Automatic scaling

Depending on the problem, an appropriate scaling of the objective function, of the constraints and of the optimised variables might be necessary. A dedicated scaling parameter exists for the objective function (SCALOBJ). For the variables and the constraints, the scaling factor can be entered via measurement files using the syntax.

SCALING <TAG> scaling_value

The scaling factors should be set in such way that the range of values obtained by multiplying the variables and constrained values by one is about 1.

STEPMX **Input** **Step for derivatives**

Relative perturbation for calculating derivatives

EPSDER **Input** **Estimate of Noise**

Estimation of the maximum noise present in the residuals and the objective function.

FOBJ **Input** **Objective Function**

The value of the objective function (to be minimized) is usually defined by a Flex code attached to the unit.

FOBJSCAL **Input** **SCALING of objective function**

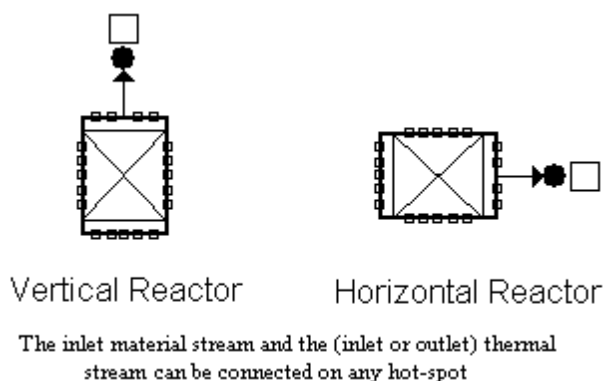
SECTION 26: RCT010 : CHEMICAL REACTOR

PURPOSE

The **RCT010** is a general-purpose reactor model where the outlet stream composition is calculated on basis of a combination of equilibrated reactions and fixed conversion reactions. A mode is also available where a White equilibrium is calculated (no reaction scheme specified). In each case, a distance from equilibrium can be considered, through a Delta T parameter. When equilibrated reactions are used, the reactor effluent **MUST** be gaseous.

The maximum number of reactions is 20, from a maximum of 10 can be equilibrated.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS 1 material stream of any state

Outlet STREAMS 1 material stream of any state, but using the same Thermod as the inlet stream.

In addition one thermal stream must be connected to the unit either as inlet or outlet stream. If the stream is connected as inlet (or outlet) stream, but heat is produced (or used), the thermal stream load will be negative.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation Mode
METH	Input	Calculation Method
KPRINT	Input	Print-out Level
TOUT	Measurable	Outlet Temperature
HEATLOAD	Measurable	Heat load
GLOBDTEQ	Measurable	Temperature deviation from equilibrium (WHITE)
DPREF	Measurable	Reference Pressure Drop
REFFLOW	Measurable	Reference Mass Flow rate
TYPE i	Input	Reaction i Type
EXTENT i	Measurable	Reaction i Extent
CONVER i	Measurable	Reaction i Conversion
CORDER i	Input	Reaction i Key compound
DTEQ i	Measurable	Reaction i DT from equilibrium

UNIT PARAMETERS

MODE	Input	Calculation Mode

0 Mass Balance only (Chemical equilibrium cannot be handled)

The outlet stream state and the thermal stream are left unchanged.

1 Set-point on Outlet Stream Temperature

The outlet temperature of the reactor is given (parameter TOUT) and the heat to be removed or to be supplied is calculated together with the outlet composition.

3 Set-point on Thermal Stream Load

The heat to be removed or to be supplied is given and the outlet temperature is calculated together with the outlet composition

METH	Input	Calculation Method

1 BROYDEN method (using given reaction schemes).

The KPs of the chemical reactions are calculated from the compounds data. The chemical compounds involved in these reactions must be gaseous and a deviation from the equilibrium may be considered for each reaction.

2 WHITE Method (ignoring given reaction scheme)

The free energy of the outlet gas, calculated on basis of compounds data, is minimised by the WHITE method. This method can only be used with gaseous chemical compounds. A unique temperature deviation from the equilibrium may be considered

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

TOUT	Measurable	Outlet Temperature

Set-point on outlet temperature (used only when Mode = 1)

HEATLOAD	Measurable	Heat load

Calculated heat load. A positive value corresponds to heat supplied to the reactor.

GLOBDTEQ	Measurable	Temperature deviation from equilibrium (WHITE)

For White Method only (if METH=2): temperature deviation from equilibrium. The equilibrium is calculated at the following temperature: TOUT + GLOBDTEQ

DPREF	Measurable	Reference Pressure Drop

REFFLOW	Measurable	Reference Mass Flow rate

DPREF < 0 No pressure drop

DPREF = 0 The pressure of outlet stream is kept

DPREF > 0 Pressure drop is calculated as follows:

REFFLOW < 0 $P_{out} = P_{in} - DPREF$

REFFLOW = 0 $P_{out} = P_{in} - DPREF$ and REFFLOW is set to the current mass flow rate (identification mode)

REFFLOW > 0 $P_{out} = P_{in} - DPREF \cdot (MASSF / REFFLOW)$

TYPE i	Input	Reaction i Type
EXTENT i	Measurable	Reaction i Extent
CONVER i	Measurable	Reaction i Conversion
CORDER i	Input	Reaction i Key compound
DTEQ i	Measurable	Reaction i DT from equilibrium

For each reaction (see Reactions page of the unit notebook), you specify the TYPE i of calculation:

Extent	the parameter EXTENT i sets the reaction extent
Conversion	the parameter CONVER i sets the conversion of the compound indicated by CORDER i
Equilibrium	the reaction is calculated at chemical equilibrium for the temperature TOUT + DTEQ i

Attention:

1. The equilibrated reactions must appear (and will be calculated) together after all reactions calculated at a given extent or conversion.
2. The reactions at given conversions or extents are calculated successively. For example, specifying 50% conversion on a given compound for the second reaction means that 50% of the remaining amount of this compound, after calculation of the first reaction will be converted in the second reaction.
3. If an extent or conversion specified on a reaction exceeds the maximum extent achievable in view of the remaining compound rates, this maximum extent will be used in place
4. For equilibrated reactions, the corresponding EXTENT i parameter will be calculated and reused as starting point in the subsequent calculations

SECTION 27: RCT110 : NH₃ SYNTHESIS REACTOR

PURPOSE

The **RCT110** is a NH₃ synthesis reactor, with four catalytic beds and inter bed quenches.

The feed of the first bed is preheated in a feed/effluent exchanger, which is integrated in the **RCT110**. Temperature control is achieved by a by-pass of the exchanger on the feed side.

The reaction is defined internally and uses the following kinetic equation:

$$r = \text{ACTIV}_j (k_1 \cdot \text{PN}_2 \cdot \text{PH}_2^{1.5} / \text{PNH}_3 - k_2 \cdot \text{PNH}_3 / \text{PH}_2^{1.5}) \quad (\text{kmol/m}^3 \text{ cata/s})$$

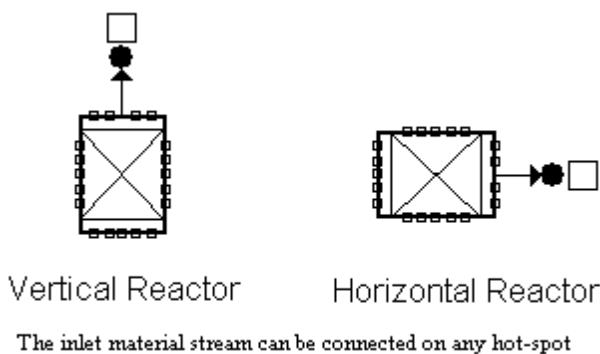
Where : $k_1 = 4.97094 \exp(-10468 / T)$

$$k_2 = 7.1428 \cdot 10^{12} \exp(-23855/T)$$

PN₂, PH₂, PNH₃ are the partial pressure of the respective compounds (bar)

ACTIV_j is the catalyst activity in the corresponding bed

SYMBOLS AND CONNECTION RULES



Inlet STREAMS 1 material stream of any state

Outlet STREAMS 1 material stream with LV state

Both streams must refer to the same Thermod, which in addition must use N₂, H₂ and NH₃ as the three first compounds.

UNIT PARAMETERS (SUMMARY)

BEDDIAM	Input	Diameter of catalytic bed
INTSTEP	Input	Integration step
HA	Input	Heat transfer factor
TIN	Input	Set-point on 1st bed inlet temperature
KPRINT	Input	Print-out Level
REFFLOW	Input	Reference mass flow rate
REFDP	Input	Reference pressure drop
EPSHA	Input	Relative precision on heat balance
FRAC _{<i>i</i>}	Measurable	Fraction of inlet flow rate going to bed <i>i</i> (see Others page)
LENGTH _{<i>i</i>}	Input	Length of catalyst bed <i>i</i>
ACTIV _{<i>i</i>}	Input	Catalyst activity in bed <i>i</i>

UNIT PARAMETERS

BEDDIAM	Input	Diameter of catalytic bed
INTSTEP	Input	Integration step
HA	Input	Heat transfer factor

Overall heat transfer factor of the feed/effluent exchanger. The heat transfer is calculated as

$$Q = HA \cdot DTLN$$

TIN	Input	Set-point on 1st bed inlet temperature

The by-pass on the feed/effluent exchanger will be controlled so as to reach the set temperature.

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

REFFLOW	Input	Reference mass flow rate
REFDP	Input	Reference pressure drop

$$P_{out} = P_{in} - DPREF \cdot (MASSF / REFFLOW)^2$$

If REFFLOW or REFDP are negative, the pressure drop will be zero.

EPSHA	Input	Relative precision on heat balance
FRAC i	Measurable	Fraction of inlet flow rate going to bed i (see Others page)
LENGTH i	Input	Length of catalyst bed i
ACTIV i	Input	Catalyst activity in bed i

These parameters are to be defined for each of the four beds.

SECTION 28: RCT141 : FURNACE TYPE CHEMICAL REACTOR

PURPOSE

The **RCT140** is a model simulating a furnace where the fluid flowing through the tubes undergoes chemical reactions.

The reactor outlet stream composition is calculated on basis of a combination of equilibrated reactions and fixed conversion reactions. A mode is also available where a White equilibrium is calculated (no reaction scheme specified). In each case, a distance from equilibrium can be considered, through a Delta T parameter.

When equilibrated equations are used, the reactor outlet must be gaseous.

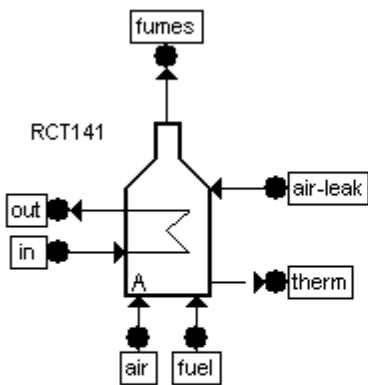
The fuel can be either a gaseous or liquid substance. Combustion is considered to be complete and the necessary air flow rate can be calculated according to the air excess or the percentage of O₂ or CO₂ in the fumes.

The heat exchange occurs by radiation. The heat transfer coefficient and the pressure drop inside the tubes is calculated through internal correlations.

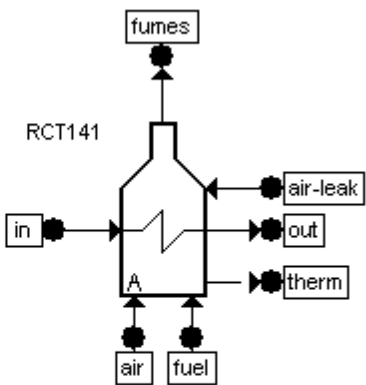
The unit can be calculated for a given fuel flow rate or for a given reactor effluent temperature.

The maximum number of reactions is 10.

SYMBOLS AND CONNECTION RULES



Chemical Furnace left/left



Chemical Furnace left/right

Inlet STREAMS	1	Air Feed: material stream bringing the oxygen
	2	Fuel Feed: material stream bringing the fuel to be burned
	3	Reactor Feed
	4	Optional air leakage stream
Outlet STREAMS	1	Fumes: material stream with combustion products
	2	Reactor Effluent (LV state), using the same Thermod as the Reactor feed
	3	Optional Thermal Stream for heat losses

The Thermod used by the Fumes streams must contain at least all the components that are present in the air and the fuel feed streams.

UNIT PARAMETERS (SUMMARY)

MODEXC	Input	Air flow calculation mode
MODCAL	Input	Furnace calculation mode
METH	Input	Reactor calculation mode
NRC	Input	Number of combustion reactions
NTUBES	Input	Number of tubes
NTUBPASS	Input	Number of passes
LTUB	Input	Tubes length
DEXT	Input	Tubes outside diameter
DINT	Input	Tubes inside diameter
COND	Input	Tubes thermal conductivity
GEOMFACT	Input	Radiation chamber geometric factor
KPRINT	Input	Print-out Level
TGOUT	Input	Set point on outlet gas temperature
GLOBDTEQ	Measurable	Delta T at equilibrium (White)
SETPOINT	Measurable	Air excess or fumes O ₂ /CO ₂ contents
CFDP	Input	Pressure drop correction factor
EPS	Input	Relative precision on heat balance
CFH	Input	Heat transfer correction factor
HLOSSF	Measurable	Heat losses (as a fraction of load)
TYPE _{<i>i</i>}	Input	Reaction <i>i</i> calculation type
EXTENT _{<i>i</i>}	Measurable	Extent of reaction <i>i</i>
CONVER _{<i>i</i>}	Measurable	Conversion of key compound
CORDER _{<i>i</i>}	Input	Key compound
DTEQ _{<i>i</i>}	Measurable	Delta T from equilibrium

UNIT PARAMETERS

MODEXC	Input	Air flow calculation mode

- 0 The air flow rate is given in the Air inlet stream
- 1 SETPOINT on the air excess
- 2 SETPOINT on the O₂ content in fumes
- 3 SETPOINT on the CO₂ content in dry fumes
- 4 SETPOINT on the CO₂ content in fumes

For MODEXC = 1.4, the flow rate of the Air inlet stream will be calculated.

MODCAL Input Furnace calculation mode

- 1 Fuel flow rate is set
The temperature reactor outlet stream is calculated from the heat balance
- 2 Reactor outlet temperature (TGOUT) is set
The fuel flow rate is calculated in order to get the specified reactor outlet temperature
In this case, the optional air leak stream cannot be used.

METH Input Reactor calculation mode

- 1 **BROYDEN method (using given reaction schemes).**
The KPs of the chemical reactions are calculated from the compounds data. The chemical compounds involved in these reactions must be gaseous and a deviation from the equilibrium may be considered for each reaction.
- 2 **WHITE Method (ignoring given reaction scheme)**
The free energy of the outlet gas, calculated on basis of compounds data, is minimised by the WHITE method. This method can only be used with gaseous chemical compounds. A unique temperature deviation from the equilibrium may be considered

NRC Input Number of combustion reactions

The corresponding first reactions are used to describe the combustion. Complete combustion is always assumed.

NTUBES	Input	Number of tubes
NTUBPASS	Input	Number of passes
LTUB	Input	Tubes length
DEXT	Input	Tubes outside diameter
DINT	Input	Tubes inside diameter

COND **Input** **Tubes thermal conductivity**

If COND <= 0, the tube thermal resistance is neglected

GEOMFACT **Input** **Radiation chamber geometric factor**

The heat exchange is controlled by radiation:

$$\text{LOAD} = \text{CFH} \cdot \text{GEOMFACT} \cdot \sigma \cdot \text{AREA} \cdot (\text{TFUMES}^4 - \text{TWALL}^4)$$

Where : σ is the Boltzmann constant (5.667 10⁻⁸ W/m² K⁴)

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

TGOUT **Input** **Set point on outlet gas temperature**

This parameter is only used when MODCAL=2.

GLOBDTEQ **Measurable** **Delta T at equilibrium (White)**

For White Method only (if METH=2): temperature deviation from equilibrium. The equilibrium is calculated at the following temperature: TOUT + GLOBDTEQ

SETPOINT **Measurable** **Air excess or fumes O2/CO2 contents**

The SETPOINT is used for controlling the air flow rate, according to the air flow calculation mode defined by the MODEXC parameter.

CFDP **Input** **Pressure drop correction factor**

This multiplying factor can be used to tune the pressure drop correlation equation to actual plant conditions. If CFPD <= 0, no correction is applied

EPS **Input** **Relative precision on heat balance**

If EPS <= 0, the default value is 10⁻⁴.

CFH **Input** **Heat transfer correction factor**

This multiplying factor can be used to tune the heat transfer correlation equation to actual plant conditions. If CFPD <= 0, no correction is applied.

HLOSSF **Measurable** **Heat losses (as a fraction of load)**

The corresponding fraction of the furnace heat load is considered as heat losses and put in the optional thermal outlet stream when present.

TYPE_i	Input	Reaction <i>i</i> calculation type
EXTENT_i	Measurable	Extent of reaction <i>i</i>
CONVER_i	Measurable	Conversion of key compound
CORDER_i	Input	Key compound
DTEQ_i	Measurable	Delta T from equilibrium

For each reaction (see Reactions page of the unit notebook), you specify the TYPE_i of calculation:

Combustion	used for the first NRC reactions. For these reactions the other parameters are not used
Extent	the parameter EXTENT _i sets the reaction extent
Conversion	the parameter CONVER _i sets the conversion of the compound indicated by CORDER _i
Equilibrium	the reaction is calculated at chemical equilibrium for the temperature TOUT + DTEQ _i

Attention:

1. The equilibrated reactions must appear (and will be calculated) together after all reactions calculated at a given extent or conversion.
2. The reactions at given conversions or extents are calculated successively. For example, specifying 50% conversion on a given compound for the second reaction means that 50% of the remaining amount of this compound, after calculation of the first reaction will be converted in the second reaction.
3. If an extent or conversion specified on a reaction exceeds the maximum extent achievable in view of the remaining compound rates, this maximum extent will be used in place
4. For equilibrated reactions, the corresponding EXTENT_i parameter will be calculated and reused as starting point in the subsequent calculations.

SECTION 29: RCT201 : CHEMICAL REACTOR

PURPOSE

Calculates the outlet composition, taking into account specifications such as :

- the conversion or extent is given for some of the reactions,
- the conversion or extent is given for some of the components,
- the user can give his own values for chemical equilibrium ratio and
- the other reactions are at usual chemical equilibrium.

Specifications are optional depending on the user's choice. By default, all the reactions are considered at chemical equilibrium.

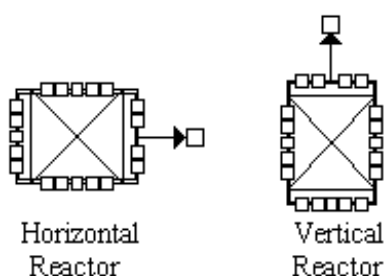
As a principle, all reactions are fluid phase reactions (liquid or vapour). However, it may be possible to consider reactions involving solids, if we have the right value for the GIBBS' free energy of formation (for the pure solid at 25 C and 1 atm). It may also be possible to consider reactions in a LV phase reactor provided ALL reactions are fixed conversion reactions. If the WHITE method is used to solve the problem, reactions MUST be in the vapour phase without solids.

The maximum number of reactions is 20.

RCT010 and **RCT201** units are similar, but differs on several aspects for the reactions at equilibrium:

- RCT010 only supports vapour phase, while RCT020 supports both full vapour and full liquid phases.
- In addition RCT210 also handles solids.
- RCT010 allows to set a different temperature deviation for each reaction while **RCT201** only allows a global temperature deviation from equilibrium.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Feed: Material Stream of any state (Liquid, Liquid/Vapour, ...)
Outlet STREAMS	1	Product: Material Stream with LV state

In addition one thermal stream can be connected to the unit either as inlet or outlet stream. If the stream is connected as inlet (or outlet) stream, but heat is produced (or used), the thermal stream load will be negative.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation Mode
METH	Input	Calculation Method
KPRINT	Input	Print-out Level
TOUT	Measurable	Outlet Temperature
HEATLOAD	Measurable	Heat load
GLOBDTEQ	Measurable	Temperature deviation from equilibrium
REFDP	Measurable	Reference Pressure Drop
REFFLOW	Measurable	Reference Mass Flow rate
TYPE i	Input	Reaction i Type
EXTENT i	Measurable	Reaction i Extent
CONVER i	Measurable	Reaction i Conversion
CORDER i	Input	Reaction i Key compound

UNIT PARAMETERS

MODE	Input	Calculation Mode
0	Mass Balance only	(Chemical equilibrium cannot be handled) The outlet stream state and the thermal stream are left unchanged.
1	Set-point on Outlet Stream Temperature	The outlet temperature of the reactor is given (parameter TOUT) and the heat to be removed or to be supplied is calculated together with the outlet composition.
3	Set-point on Thermal Stream Load	The heat to be removed or to be supplied is given and the outlet temperature is calculated together with the outlet composition

METH	Input	Calculation Method

1 BROYDEN method (using given reaction schemes).

The KPs of the chemical reactions are calculated from the compounds data. The chemical compounds involved in these reactions must be gaseous and a deviation from the equilibrium may be considered for each reaction.

2 WHITE Method (ignoring given reaction scheme)

The free energy of the outlet gas, calculated on basis of compounds data, is minimised by the WHITE method. This method can only be used with gaseous chemical compounds. A unique temperature deviation from the equilibrium may be considered

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

TOUT **Measurable** **Outlet Temperature**

Set-point on outlet temperature (used only when Mode = 1)

HEATLOAD **Measurable** **Heat load**

Calculated heat load. A positive value corresponds to heat supplied to the reactor.

GLOBDTEQ **Measurable** **Temperature deviation from equilibrium**

Temperature deviation from equilibrium. The equilibrium is calculated at the following temperature: TOUT + GLOBDTEQ

REFDP **Measurable** **Reference Pressure Drop**

REFFLOW **Measurable** **Reference Mass Flow rate**

DPREF < 0 No pressure drop

DPREF = 0 The pressure of outlet stream is kept

DPREF < 0 Pressure drop is calculated as follows:

REFFLOW < 0 $P_{out} = P_{in} - DPREF$

REFFLOW = 0 $P_{out} = P_{in} - DPREF$ and REFFLOW is set to the current mass flow rate (identification mode)

REFFLOW > 0 $P_{out} = P_{in} - DPREF \cdot (MASSF / REFFLOW)$

TYPE <i>i</i>	Input	Reaction <i>i</i> Type
EXTENT <i>i</i>	Measurable	Reaction <i>i</i> Extent
CONVER <i>i</i>	Measurable	Reaction <i>i</i> Conversion
CORDER <i>i</i>	Input	Reaction <i>i</i> Key compound

For each reaction (see Reactions page of the unit notebook), you specify the TYPE*i* of calculation:

Extent	the parameter EXTENT <i>i</i> sets the reaction extent
Conversion	the parameter CONVER <i>i</i> sets the conversion of the compound indicated by CORDER <i>i</i>
Equilibrium	the reaction is calculated at chemical equilibrium for the temperature TOUT + DTEQ, the KPs of the reactions being calculated from the compounds data
Kp	the reaction is calculated at chemical equilibrium for the temperature TOUT + DTEQ, the KPs of the reactions being calculated from the parameters KA and KB of the reaction, as given by the user: $\ln(K) = KA - KB/R/T$.

Attention:

1. The equilibrated reactions must appear (and will be calculated) together after all reactions calculated at a given extent or conversion.
2. The reactions at given conversions or extents are calculated successively. For example, specifying 50% conversion on a given compound for the second reaction means that 50% of the remaining amount of this compound, after calculation of the first reaction will be converted in the second reaction.
3. If an extent or conversion specified on a reaction exceeds the maximum extent achievable in view of the remaining compound rates, this maximum extent will be used in place
4. For equilibrated reactions, the corresponding EXTENT*i* parameter will be calculated and reused as starting point in the subsequent calculations
5. The compounds selected as reference for the conversions must be reagents (negative stoichiometric coefficients in the reaction).
6. It is not possible to impose a conversion or an extent of a component whose inlet flowrate is 0, even it appears during the reaction.

SECTION 30: RCT210 : CHEMICAL REACTOR

PURPOSE

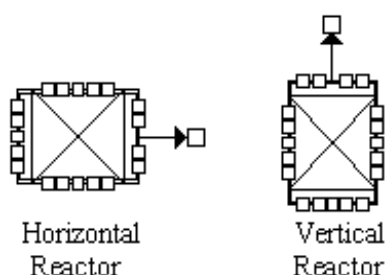
Calculates the outlet composition on basis of White method, by minimising the free energy of the outlet stream at an equilibrium temperature calculated from the actual stream temperature and a temperature deviation (DTEQ).

The reactions occur in vapour phase. Solid compounds can be taken into account.

The reactor can be calculated at a given outlet temperature or at a given heat input or output.

The maximum number of reactions is 20.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Feed: Material Stream of any state (Liquid, Liquid/Vapour, ...)

Outlet STREAMS	1	Product: Material Stream with LV state

In addition one thermal stream can be connected to the unit either as inlet or outlet stream. If the stream is connected as inlet (or outlet) stream, but heat is produced (or used), the thermal stream load will be negative.

UNIT PARAMETERS (SUMMARY)

KPRINT	Input	Print-out Level
MODE	Input	Calculation Mode
TOUT	Measurable	Outlet Temperature
HEATLOAD	Measurable	Heat load
GLOBDTEQ	Measurable	Temperature deviation from equilibrium
REFDP	Measurable	Reference Pressure Drop
REFFLOW	Measurable	Reference Mass Flow rate
TYPE i	Input	Reaction i Type
EXTENT i	Measurable	Reaction i Extent
CONVER i	Measurable	Reaction i Conversion
CORDER i	Input	Reaction i Key compound

UNIT PARAMETERS

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

MODE	Input	Calculation Mode

1 Set-point on Outlet Stream Temperature

The outlet temperature of the reactor is given (parameter TOUT) and the heat to be removed or to be supplied is calculated together with the outlet composition.

2 Set-point on HeatLoad parameter

The heat to be removed or to be supplied is given in the parameter HEATLOAD and the outlet temperature is calculated together with the outlet composition

3 Set-point on Thermal Stream Load

The heat to be removed or to be supplied is given in the thermal stream and the outlet temperature is calculated together with the outlet composition

TOUT	Measurable	Outlet Temperature

Set-point on outlet temperature (used only when Mode = 1)

HEATLOAD	Measurable	Heat load

Calculated or specified (MODE=2) heat load. A positive value corresponds to heat supplied to the reactor.

GLOBDTEQ Measurable Temperature deviation from equilibrium

Temperature deviation from equilibrium. The equilibrium is calculated at the following temperature: TOUT + GLOBDTEQ

REFDP Measurable Reference Pressure Drop

REFFLOW Measurable Reference Mass Flow rate

DPREF < 0 No pressure drop

DPREF = 0 The pressure of outlet stream is kept

DPREF < 0 Pressure drop is calculated as follows:

REFFLOW <=0 $P_{out} = P_{in} - DPREF$ and REFFLOW is set to the current mass flow rate (identification mode)

REFFLOW > 0 $P_{out} = P_{in} - DPREF \cdot (MASSF/REFFLOW)^2$

TYPE_i Input Reaction *i* Type

EXTENT_i Measurable Reaction *i* Extent

CONVER_i Measurable Reaction *i* Conversion

CORDER_i Input Reaction *i* Key compound

For each reaction (see Reactions page of the unit notebook), you specify the TYPE_i of calculation:

Extent the parameter EXTENT_i sets the reaction extent

Conversion the parameter CONVER_i sets the conversion of the compound indicated by CORDER_i

Equilibrium the reaction is calculated at chemical equilibrium for the temperature TOUT + DTEQ, the KPs of the reactions being calculated from the compounds data

Kp the reaction is calculated at chemical equilibrium for the temperature TOUT + DTEQ, the KPs of the reactions being calculated from the parameters KA and KB of the reaction, as given by the user: $\ln(K) = KA - KB/R/T$.

Attention:

- The equilibrated reactions must appear (and will be calculated) together after all reactions calculated at a given extent or conversion.**
- The reactions at given conversions or extents are calculated successively. For example, specifying 50% conversion on a given compound for the second reaction means that 50% of the remaining amount of this compound, after calculation of the first reaction will be converted in the second reaction.**

3. If an extent or conversion specified on a reaction exceeds the maximum extent achievable in view of the remaining compound rates, this maximum extent will be used in place
4. For equilibrated reactions, the corresponding EXTENT_i parameter will be calculated and reused as starting point in the subsequent calculations
5. The compounds selected as reference for the conversions must be reagents (negative stoichiometric coefficients in the reaction).
6. It is not possible to impose a conversion or an extent of a component whose inlet flowrate is 0, even it appears during the reaction.

SECTION 31: RCT300 : INTEGRAL PLUG FLOW REACTOR

PURPOSE

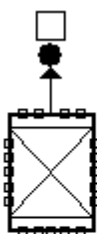
UNIT **RCT300** is a simulation model for a chemical integral plug flow reactor, and can be used in simulation (reactor volume is given) or sizing mode (the reactor volume is calculated

Homogeneous and heterogeneous catalytic reactors as well as non-catalytic reactors can be modelled. However, the feed stream must be in homogeneous phase (fully liquid or fully vapour) along the reactor. Should a phase change occur, the simulation will be cancelled.

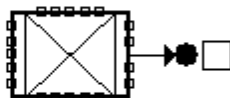
Up to 21 reactions can be defined, while the reactor itself can be divided in up to 5 zones.

Reactions defined in Reactions page + Flex kinetics +

SYMBOLS AND CONNECTION RULES



Vertical Reactor



Horizontal Reactor

The inlet stream can be connected to any hot spot

Inlet STREAMS	1	Material inlet stream (any state)
Outlet stream	1	Material outlet stream (LV state), using the same Thermod as the inlet stream.

UNIT PARAMETERS (SUMMARY)

KPRINT	Input	Print-out Level
BETA	Input	Catalyst density (kmol/m ³ or kg/m ³) (1 if none)
NZONE	Input	Number of zones
MODE	Input	Calculation mode
REFCOMP	Input	Reference compound
CONVER	Input	Conversion of reference compound
REFDP	Measurable	Reference pressure drop
REFFLOW	Measurable	Reference mass flow rate
TYPZO _{<i>i</i>}	Input	Type of heat transfer in Zone <i>i</i>
REVOL _{<i>i</i>}	Measurable	Zone volume
HA _{<i>i</i>}	Input	Zone HA
TINZO _{<i>i</i>}	Input	Zone inlet temperature
DTZO _{<i>i</i>}	Input	Zone temperature drop
INTSTEP	Input	Initial value of integration step
EPSTIF	Input	Tolerance for integration routine
EPSHEAT	Input	Precision on heat balance
ITMHEAT	Input	Maximum iterations on heat balance
EPSVR	Input	Precision on reactor volume
ITMVR	Input	Maximum iterations on reactor volume
EPSVAPF	Input	Tolerance on vapour or liquid fraction
PROFILE	Input	Number of steps for profile report
EPSMB	Input	Precision on mass balance

UNIT PARAMETERS

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

0 - 1	No printing
2	Reminder for heat transfer mode
3	Iterations on reactor volume
4	Iterations on thermal balance
5	Intermediate integration values

BETA	Input	Catalyst density (kmol/m³ or kg/m³) (1 if none)

The signification of BETA depends on the type of reactor:

Homogeneous reactor: BETA = 1

Catalytic reactor in homogeneous phase: BETA is expressed in kmol cata/m³

Catalytic reactor in heterogeneous phase: BETA is expressed in kg cata/m³

The material balance in each reactor zone is calculated as:

$$dU_i/dl = BETA \cdot VOLUME \cdot r_i$$

Where :

U _i	is the extent of reaction i (kmole/s)
l	is the position in the reactor zone (from 0 to 1)
VOLUME	is the zone volume (m ³)
r _i	is the reaction rate, which signification is also depending on the type of reactor:

Homogeneous reactor: r_i is expressed in kmol/s

Catalytic reactor in homogeneous phase: r_i is expressed in kmol/(s.kmol cata)

Catalytic reactor in heterogeneous phase: r_i is expressed in kmol/(s.kg cata)

The reaction rate r_i is expressed as a product of two factors:

$$r_i = V_i \cdot E_i$$

Where :

V _i	is the reaction velocity (in kmole/s or kmol/(s.kmol cata) or kmol/(s.kg cata))
E _i	is an equilibrium factor equal to 0 when equilibrium is reached

The way V_i and E_i are calculated is determined in the Parameters page of the Reaction notebook.

Attention:

1. As a general rule, all the reactions should be defined on basis of the same type of equilibrium and velocity equations.
2. However, it is allowed to use the equilibrium equation of type "8" (E_i = 1) for any reaction
3. When the reaction velocity boils down to a constant (when the overall order is 0) the equilibrium equation must be set to "8" (E_i = 1).
4. The fugacity coefficients (when used) are calculated on basis of the conditions at the inlet of each zone.

NZONE	Input	Number of zones

The reactor may be split into up to 5 of zones.

MODE	Input	Calculation mode
1	Simulation mode:	the reactor volume (in up to 5 zones) is given.
2	Sizing mode:	the reactor volume (1 zone maximum) is calculated in order to reach a given conversion for a reference compound.

REFCOMP	Input	Reference compound
CONVER	Input	Conversion of reference compound

These parameters are only used in **Sizing mode** and indicate the conversion of a reference compound.

REFDP	Measurable	Reference pressure drop
REFFLOW	Measurable	Reference mass flow rate

The pressure drop is expressed as:

$$DP = REFDP \cdot (MASSF / REFFLOW)^2$$

Where : G is the mass flow rate.

TYPZO _i	Input	Type of heat transfer in Zone <i>i</i>
REVOL _i	Measurable	Zone volume
HA _i	Input	Zone HA
TINZO _i	Input	Zone inlet temperature
DTZO _i	Input	Zone temperature drop

These parameters must be defined for each zone.

TYPZO _i =0:	Not used	This zone is not used
TYPZO _i =1:	T.Profile	The temperature profile of the zone is given by

$$T_i = TINZO_i + DTZO_i \cdot L$$

Where : L is the normalised length for each zone, and goes from 0 to 1.

Assuming a heat transfer towards or from an isothermal environment at temperature TR, the heat exchanged with this isothermal environment is computed by the program.

$$\begin{aligned} Q > 0 \text{ heat supply,} & \quad Q = -HA \cdot (\text{Max}(TO_i, TO_i + DTZO_i) - TR) \\ Q < 0 \text{ heat removal,} & \quad Q = -HA \cdot (\text{Min}(TO_i, TO_i + DTZO_i) - TR) \end{aligned}$$

TYPZO_i=2: **Adiabatic** No heat transfer in the zone

INTSTEP **Input** **Initial value of integration step**

Initial value of the integration step, expressed as a fraction of reactor length.

If PROFILE = 0, then the value 1/20 is used.

If PROFILE ≤ 0, then the default attribute 1/PROFILE is taken.

EPSTIF **Input** **Tolerance for integration routine**

EPSHEAT **Input** **Precision on heat balance**

ITMHEAT **Input** **Maximum iterations on heat balance**

EPSHEAT **Input** **Precision on heat balance**

ITMHEAT **Input** **Maximum iterations on heat balance**

The parameter EPSVR and ITMV are used only in sizing mode.

EPSVAPF **Input** **Tolerance on vapour or liquid fraction**

Deviation from 0 or 1 on vapour fraction for a phase to be regarded as homogeneous.

If EPSVAPF < 0, then the program chooses the default value: 10⁻⁴.

The maximum value for this parameter is 0.05.

PROFILE **Input** **Number of steps for profile report**

Number of intermediary values for profiles printed in report phase.

If PROFILE ≤ 0, then the program chooses the default attribute of 10.

The maximum value for this parameter is 100.

EPSMB **Input** **Precision on mass balance**

SECTION 32: RCT320 : COOLED PLUG FLOW REACTOR

PURPOSE

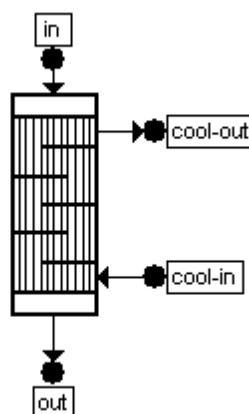
UNIT **RCT320** is a simulation model for a chemical integral plug flow reactor with detailed modelling of a coolant media. The reactor can be split up in three zones with different catalyst activities.

Homogeneous and heterogeneous catalytic reactors as well as non-catalytic reactors can be modelled. However, the feed stream must be in homogeneous phase (fully liquid or fully vapour) along the reactor. Should a phase change occur, the simulation will be cancelled.

Up to 21 reactions can be defined.

Reactions defined in Reactions page + Flex kinetics +

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Material inlet stream (any state)
	2	Inlet coolant stream
Outlet STREAMS	1	Material outlet stream (LV state), using the same Thermod as the first inlet stream.
	2	Outlet coolant stream using the same Thermod as the inlet coolant stream

UNIT PARAMETERS (SUMMARY)

EXCHMODE	Input	Cooling flow configuration
TUBLAYOU	Input	Tubes Lay-out
DISPHV	Input	Position of device
ISAVIN	Input	Print profiles file
FILEFORM	Input	Profiles File Format
KPRINT	Input	Print-out Level
BETA	Input	Correction factor on Reaction Velocity
REFCOMP	Input	Reference Compound for Conversion
REFPROD	Input	Reference Compound for Selectivity
REFDP	Input	Reference Pressure Drop
REFFLOW	Input	Reference Mass Flow rate
DCAT	Input	Catalyst Pellets Diameter
NTUBES	Input	Number of Tubes
NTUBPASS	Input	Number of Passes (Tube side)
LENGTH	Input	Tubes Length
TD	Input	Tubes Outside Diameter
TDIN	Input	Tubes Inside Diameter
COND	Input	Tubes Thermal Conductivity
TUBPITCH	Input	Tubes Spacing
BUNDLDIA	Input	Bundle Diameter
SHELLDIA	Input	Shell Diameter
NBAFFLES	Input	Number of Baffles
BAFFOPEN	Input	Baffles opening
FF	Input	Fouling Factor
CFH	Input	Heat Transfer Correction factor
INTSTEP	Input	Integration step
EPSTIF	Input	Precision required for integration
EPSHEAT	Input	Precision on counter-current transfer
ITMHEAT	Input	Maximum Number of Iterations
EPSVAPF	Input	Precision on Vapour Fraction
NSTEPS	Input	Number of values for Profiles
DTMAX	Input	Maximum Temperature Difference
FRAC _j	Input	Catalyst Fraction in zone j ($j=1,3$)
ACTIV _j	Input	Catalyst Activity in zone j ($j=1,3$)
REFCOMP	Input	Reference Compound for Conversion
REFPROD	Input	Reference Compound for Selectivity

UNIT PARAMETERS

EXCHMODE Input Cooling flow configuration

EXCHMODE = 0 Co-Current

EXCHMODE = 1 Counter-Current

TUBLAYOU Input Tubes Lay-out

TUBLAYOU = 1 - Pitch type = square

TUBLAYOU = 2 - Pitch type = triangular

TUBLAYOU = 3 - Pitch type = square rotated

TUBLAYOU = 4 - Pitch type = triangle with cleaning lanes

DISPHV Input Position of device

DISPHV = 0 horizontal.

DISPHV = 1 vertical.

ISAVIN Input Print profiles file

Controls whether a separate file (named from the unit name and the extension grf) is created with the reactor profile (T, P, and molar flow rates). The number of steps is controlled by the parameter NSTEPS

ISAVIN = 0 No.

ISAVIN = 1 Yes.

FILEFORM Input Profiles File Format

Defines the field separator for the Profiles file.

FILEFORM = 32 space separator

FILEFORM = 9 TAB separator

FILEFORM = 59 semi colon separator

KPRINT Input Print-out Level

Controls the amount of information printed in the trace file.

0 - 1	No printing
2	Reminder for heat transfer mode
3	Iterations on reactor volume
4	Iterations on thermal balance
5	Intermediary integration values

BETA Input Correction factor on Reaction Velocity

The signification of BETA depends on the type of reactor:

Homogeneous reactor:	BETA = 1
Catalytic reactor in homogeneous phase:	BETA is expressed in kmol cata/m ³
Catalytic reactor in heterogeneous phase:	BETA is expressed in kg cata/m ³

The material balance in each reactor zone is calculated as:

$$dU_i/dl = BETA \cdot VOLUME \cdot r_i \cdot ACTIV_j$$

Where :	U_i	is the extent of reaction i (kmole/s)
	l	is the position in the reactor zone (from 0 to 1)
	VOLUME	is the zone volume (m ³)
	ACTIV _j	is the catalyst activity factor in zone j.
	r_i	is the reaction rate, which signification is also depending on the type of reactor:

Homogeneous reactor:	r_i is expressed in kmol/s
Catalytic reactor in homogeneous phase:	r_i is expressed in kmol/(s.kmol cata)
Catalytic reactor in heterogeneous phase:	r_i is expressed in kmol/(s.kg cata)

The reaction rate r_i is expressed as a product of two factors:

$$r_i = V_i \cdot E_i$$

Where :	V_i	is the reaction velocity (in kmole/s or kmol/(s.kmol cata) or kmol/(s.kg cata))
	E_i	is an equilibrium factor equal to 0 when equilibrium is reached

The way V_i and E_i are calculated is determined in the Parameters page of the Reaction notebook.

Attention:

- As a general rule, all the reactions should be defined on basis of the same type of equilibrium and velocity equations.

2. However, it is allowed to use the equilibrium equation of type "8" ($E_i = 1$) for any reaction
3. When the reaction velocity boils down to a constant (when the overall order is 0) the equilibrium equation must be set to "8" ($E_i = 1$).
4. The fugacity coefficients (when used) are calculated on basis of the conditions at the inlet of each zone.

REFDP	Input	Reference Pressure Drop
REFFLOW	Input	Reference Mass Flow rate

The pressure drop is expressed as:

$$DP = \text{REFDP} \cdot (\text{MASSF} / \text{REFFLOW})^2$$

Where : MASSF is the mass flow rate.

DCAT	Input	Catalyst Pellets Diameter
NTUBES	Input	Number of Tubes
NTUBPASS	Input	Number of Passes (Tube side)
LENGTH	Input	Tubes Length
TD	Input	Tubes Outside Diameter
TDIN	Input	Tubes Inside Diameter
COND	Input	Tubes Thermal Conductivity
TUBPITCH	Input	Tubes Spacing
BUNDLDIA	Input	Bundle Diameter
SHELLDIA	Input	Shell Diameter
NBAFFLES	Input	Number of Baffles
BAFFOPEN	Input	Baffles opening
FF	Input	Fouling Factor

The fouling factor enables you to account for the presence of adhering film or scale onto the surfaces. This value may be negative.

CFH	Input	Heat Transfer Correction factor

If $CFH < 0$, then 1 is chosen as the default value.

$$1/U = FF + 1/(CFH \cdot U_c)$$

Where: U_i heat transfer coefficient for the considered section.

U_c heat transfer coefficient for clean exchanger computed from correlation ($\text{kW}/(\text{K.m}^2)$).

INTSTEP **Input** **Integration step**

Initial value of the integration step, expressed as a fraction of reactor length.

If PROFILE = 0, then the value 1/20 is used.

If PROFILE <= 0, then the default attribute 1/PROFILE is taken.

EPSTIF **Input** **Precision required for integration**

EPSHEAT **Input** **Precision on counter-current transfer**

ITMHEAT **Input** **Maximum Number of Iterations**

EPSVAPF **Input** **Precision on Vapour Fraction**

Deviation from 0 or 1 on vapour fraction for a phase to be regarded as homogeneous.

If EPSVAPF < 0, then the program chooses the default value: 10⁻⁴.

The maximum value for this parameter is 0.05.

NSTEPS **Input** **Number of values for Profiles**

DTMAX **Input** **Maximum Temperature Difference**

This parameter is used in the external convergence loop on the coolant outside temperature, as an indicator of inadequate starting point. Should the temperature difference between the coolant and the reacting mixture be higher than this value, a new loop is then restarted with a lower outside temperature

FRAC_j **Input** **Catalyst Fraction in zone j ($j=1,3$)**

ACTIV_j **Input** **Catalyst Activity in zone j ($j=1,3$)**

Fraction of catalyst and corresponding relative activity for each of the maximum three zones.

REFCOMP **Input** **Reference Compound for Conversion**

REFPROD **Input** **Reference Compound for Selectivity**

The corresponding will be used as reference compound to calculate reactor conversion and selectivity.

SECTION 33: RCT400 : FLUID BED REACTOR

PURPOSE

This unit is obsolete, use RCT420 in place.

SECTION 34: RCT420 : FLUID BED REACTOR

PURPOSE

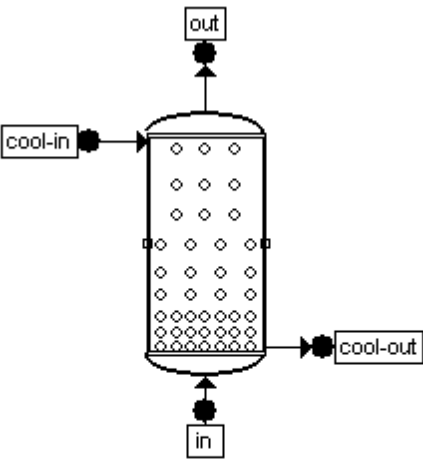
The unit **RCT420** models a fluid bed reactor. The feed stream must be in vapour state at the inlet of the reactor.

Various cooling modes are possible: adiabatic, cooling through horizontal or vertical tubes or through the external wall. When present this cooling media can in co-current or counter-current flow.

The flow pattern is determined from the inlet stream conditions and is not recalculated for the varying conditions in the reactor.

Reactions defined in Reactions page + Flex kinetics +

SYMBOLS AND CONNECTION RULES



Optional Connection:

- *Thermal Stream*

Inlet STREAMS	1	Material inlet stream (LV or V state)
	2	Inlet coolant stream (LV, L or V state)
Outlet STREAMS	1	Material outlet stream (LV or V state), using the same Thermod as the first inlet stream.
	2	Outlet coolant stream using the same Thermod as the inlet coolant stream (LV, L or V state)
	3	A thermal stream for the heat losses of the reactor

When the reactor is in adiabatic mode then only the first inlet and first outlet streams can be connected.

UNIT PARAMETERS (SUMMARY)

The information contained in this document is proprietary to BELSIM s.a. The information in this document is not to be reproduced or disclosed to third parties without the prior written consent of BELSIM s.a. This document is provided to you under a special license agreement.

UNIT PARAMETERS

FUNCMODE Input

Calculation Mode

- 1 **Pseudo-homogeneous:** the bed is considered as one homogeneous phase
- 2 **Two phase model:** two phases are considered in the bed: the dense phase and the fluid phase

FLOWMODE Input

Reactor Flow Configuration

- 1 **Perfectly Mixed Flow (PMF)**
- 2 **Plug Flow**

HEATMODE Input

Heat Transfer Mode

- 1 **Adiabatic reactor**
- 2 **Through wall of the vessel**
- 3 **Through horizontal tubes**
- 4 **Through vertical tubes**

EXCHMODE Input

Cooling Flow Configuration

- 1 **Counter-Current exchange:** an external convergence loop is handled to determine the outlet coolant stream temperature
- 2 **Co-Current exchange**

REACMODE Input

Reacting phases

- 1 **Only heterogeneous reactions**
- 2 **Heterogeneous and homogeneous reactions**

INIT

Input

Initialisation mode

- 1 **Initialisation from outlet stream**
- 2 **Initialisation from TCATIN**

DISTTYP

Input

Gas distributor type

- 1 Multiple orifices or tuyere type distributor
- 2 Porous Plate

COOLTYP Input Cooling or heating media

- 1 Cooling media
- 2 Heating media

HETEREAC Input Number of heterogeneous reactions

DCAT Input Catalyst particles diameter

FCATA Input Fraction of catalyst < 45 µm

DI5WT% Input Diameter of the 5wt% smallest cut

CATAMA Input Catalyst inventory

ROCATA Input Catalyst density

CPCATA Input Catalyst specific heat

BEDDIA Input Bed diameter

WALLTHIC Input Wall thickness

WALLCOND Input Wall thermal conductivity

NTUBES Input Number of tubes

TLENGTH Input Tubes length

TD Input Tubes diameter

TT Input Tubes Thickness

TUBPITCH Input Tubes pitch

TCOND Input Tubes thermal conductivity

BEDCORR	Input	Bed side correction coefficient.
COOLCORR	Input	Cooling side correction coefficient
REFH	Input	Cooling side reference heat transfer coefficient
REFDP	Measurable	Cooling side reference Pressure drop
REFFLOW	Measurable	Reference mass flow rate
REFDENS	Measurable	Reference density
HEXP	Input	Exponent for H equation
DPEXP	Input	Exponent for DP equation
TCATIN	Input	Initial T of the bed when PMF
KPRINT	Input	Print-out level

Controls the amount of information printed in the trace file.

DISTAO	Input	Distributor area per orifice
DIFFG	Input	Molecular diffusion coefficient
DTMAX	Input	Max. Temperature difference
EPSHEAT	Input	Precision on heat balance (Counter-current)
TMIN	Input	Minimum coolant temperature
TMAX	Input	Maximum coolant temperature

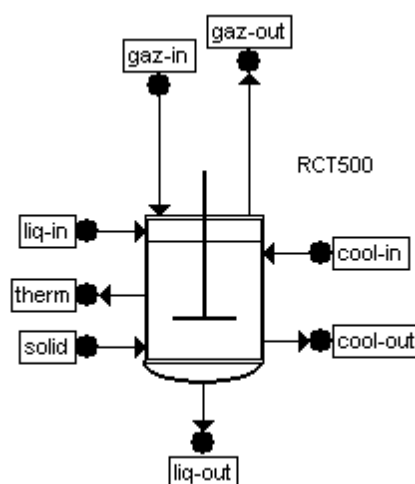
SECTION 35: RCT500 : CSTR REACTOR

PURPOSE

The unit **RCT500** models a perfectly mixed (continuous stirred) reactor.

Reactions defined in Reactions page + Flex kinetics +

SYMBOLS AND CONNECTION RULES



Inlet streams:	1	Vapour material stream (LV or V state)
	2	Liquid material stream (LV or L state)
	3	Inlet coolant stream
	4	Solid recycle stream

Outlet streams:	1	Liquid product
	2	Vapour product
	3	Outlet coolant stream
	4	Outlet Thermal stream (heat losses)

The Thermod of the liquid outlet stream must contain all compounds, including solids, if any.

The Themod of the vapour outlet stream must contain all fluid compounds. In addition it can also contain the solid compounds if any.

These two Thermods must use the same thermodynamic methods.

The inlet and outlet coolant streams must use the same Thermod.

The heat of reaction of each reaction must be defined in the reaction parameters (parameters DHR and REFT).

UNIT PARAMETERS (SUMMARY)

MODE	Input	Cooling type
DIREAC	Input	Reactor Diameter
CATAMA	Input	Catalyst Inventory
ROCATA	Input	Catalyst Density
HREAC	Input	Reactor Height
DCAT	Input	Catalyst Pellets Diameter
FF	Input	Tube Fouling Factor
KPRINT	Input	Print-out Level
TD	Input	Cooling Tubes Diameter
TT	Input	Cooling Tubes Thickness
TCOND	Input	Cooling Tubes Thermal Conductivity
NTUBES	Input	Number of Cooling Tubes
LENGTH	Input	Cooling Tubes Length
LVAREA	Input	Liquid/Vapour Heat Transfer Area
HWALL	Input	Height of Cooled Reactor Wall
REFH	Input	Heat Transfer Coefficient (Tube side)
REFDP	Input	Reference Pressure Drop
REFFLOW	Input	Reference Mass Flow rate
REFDENS	Input	Reference Density
HEXP	Input	Exponent for Tube side Heat Transfer
DPEXP	Input	Exponent for Pressure Drop
HLS	Input	Heat Transfer Coef. (liquid/solid)
TCATIN	Input	Estimation of Catalyst Temperature
HSGASGAS	Input	Gas/Gas Diffusion Coef.(L/S interface)
HSGASLIQ	Input	Liq/Gas Diffusion Coef.(L/S interface)
HSLIQLIQ	Input	Liq/Liq Diffusion Coef.(L/S interface)
HLGASGAS	Input	Gas/Gas Diffusion Coef.(Liquid phase)
HLGASLIQ	Input	Liq/Gas Diffusion Coef. (Liquid phase)
HLLIQLIQ	Input	Liq/Liq Diffusion Coef.(Liquid phase)
HVGASGAS	Input	Gas/Gas Diffusion Coef.(Gas phase)
HVGASLIQ	Input	Liq/Gas Diffusion Coef.(Gas phase)
HVLIQLIQ	Input	Liq/Liq Diffusion Coef.(Gas phase)

UNIT PARAMETERS

MODE	Input	Cooling type

1 **Tubes:** the reactor is cooled down by a coolant flowing in tubes

2 **External blanket:** the reactor is cooled down by a coolant flowing in an external blanket

DIREAC	Input	Reactor Diameter
CATAMA	Input	Catalyst Inventory
ROCATA	Input	Catalyst Density
HREAC	Input	Reactor Height
DCAT	Input	Catalyst Pellets Diameter
FF	Input	Tube Fouling Factor

KPRINT	Input	Print-out Level (0 - 8)

Controls the amount of information printed in the trace file.

TD	Input	Cooling Tubes Diameter
TT	Input	Cooling Tubes Thickness
TCOND	Input	Cooling Tubes Thermal Conductivity
NTUBES	Input	Number of Cooling Tubes
LENGTH	Input	Cooling Tubes Length

These parameters are only used when MODE=1 (Coolant flowing through tubes).

LVAREA	Input	Liquid/Vapour Heat Transfer Area
HWALL	Input	Height of Cooled Reactor Wall

These parameters are only used when MODE=2 (Coolant flowing through an external blanket).

REFH	Input	Heat Transfer Coefficient (Tube side)
REFDP	Input	Reference Pressure Drop
REFFLOW	Input	Reference Mass Flow rate
REFDENS	Input	Reference Density
HEXP	Input	Exponent for Tube side Heat Transfer
DPEXP	Input	Exponent for Pressure Drop

The pressure drop is expressed as:

REFDP > 0 DP = REFDP . (MASSF/REFFLOW)**DPEXP * REFDENS/DENS

REFDP <= 0 DP = 0

The heat transfer coefficient on the tube side is calculated as follows:

$$H = \text{REFH} * (\text{MASSF} / \text{REFFLOW})^{(\text{HEXP})}$$

When the reference terms (REFFLOW or REFDENS) are negative or zero, then the corresponding dependence is ignored.

HLS	Input	Heat Transfer Coef. (liquid/solid)
TCATIN	Input	Estimation of Catalyst Temperature
HSGASGAS	Input	Gas/Gas Diffusion Coef.(L/S interface)
HSGASLIQ	Input	Liq/Gas Diffusion Coef.(L/S interface)
HSLIQLIQ	Input	Liq/Liq Diffusion Coef.(L/S interface)
HLGASGAS	Input	Gas/Gas Diffusion Coef.(Liquid phase)
HLGASLIQ	Input	Liq/Gas Diffusion Coef. (Liquid phase)
HLLIQLIQ	Input	Liq/Liq Diffusion Coef.(Liquid phase)
HVGASGAS	Input	Gas/Gas Diffusion Coef.(Gas phase)
HVGASLIQ	Input	Liq/Gas Diffusion Coef.(Gas phase)
HVLIQLIQ	Input	Liq/Liq Diffusion Coef.(Gas phase)

SECTION 36: REPORT : REPORT FILE CONTROLLER

PURPOSE

The **REPORT** unit enables to print-out the value of up to 10 variables at each execution of the unit in an ASCII file (named from the unit name and the extension **rep**).

This unit can for instance be used in combination with an INCREM unit for a parametric sensitivity analysis on a model. It can also be used as a debugging tool in convergence loops.

The first two lines of the report file display the names and the physical units of the variables. The following lines (1 line per iteration) contain the values of the selected variables. This file can be imported into Excel for further processing like charts generation.

SYMBOLS AND CONNECTION RULES



You cannot connect any stream to an INCREM unit.

UNIT PARAMETERS

This UNIT has no parameter.

SPECIFYING THE VARIABLES TO BE PRINTED

The variables to be printed are defined via the **Control Tags** page of the Unit Notebook. The list of existing Tags is shown in the **Tags Defined** list. You can use the **Mask** field to limit the number of tags displayed in this list.

The screenshot shows a software window titled "Name: RP-1/SEQ1 Category: Controller Type: REPORT". It has a tabbed interface with the following tabs: General, Parameters, Measurements, Control Tags (selected), Reactions, Others, and Flexcode. The "Control Tags" tab contains two list boxes. The left list box contains the text "W-1_POWER", "2_P", and "2_T". The right list box is titled "Tags Defined" and is currently empty. Between the two list boxes are two buttons labeled "Up" and "Down". At the bottom of the window, there is a "Mask" label followed by an empty text input field. Below the input field are three buttons: "Ok", "Cancel", and "Help".

Name: RP-1/SEQ1 Category: Controller Type: REPORT

General Parameters Measurements Control Tags Reactions Others Flexcode

W-1_POWER
2_P
2_T

Tags Defined

Up
Down

Mask

Ok Cancel Help

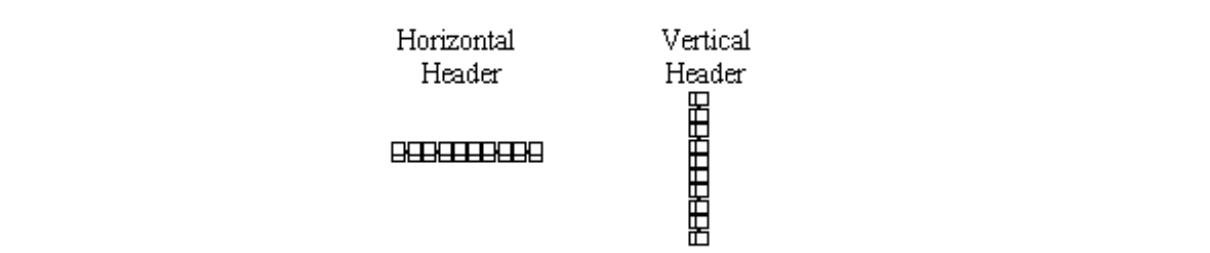
SECTION 37: SEP010 : SPLITTER

PURPOSE

Splits a mixture into a maximum of 10 outlet STREAMS, according to the user specified split fractions. Temperatures, pressure and the compositions of the outlet streams are identical to those of the inlets.

This type of UNIT may also be used for mass balance calculations. In this case the contents of the outlet streams remain unchanged.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS : 1 material stream (any state)

Outlet STREAMS : From 1 to 10 material streams (any state)

All streams must use the same Thermod.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation mode
FRAC _{<i>i</i>}	Measurable	Split fraction of stream <i>i</i>

UNIT PARAMETERS

MODE	Input	Calculation mode

0 **Heat and mass balance**

The outlet streams are set in the same state as the inlet stream.

1 **Mass balance only**

The enthalpy of the outlet streams is set to -1.

FRACi	Measurable	Split fraction of stream i

If all FRAC i are defined (≥ 0), their sum should be equal to 1.

If only one FRAC i is undefined (< 0), it is calculated by difference (1-sum of defined fractions). In this case, the sum of defined fractions has to be lower than 1.

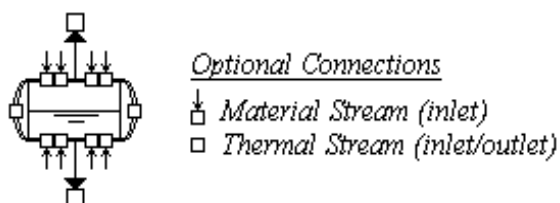
SECTION 38: SEP020 : LIQUID - VAPOUR SEPARATOR

PURPOSE

This UNIT calculates mass and energy balances of a liquid-vapour separator, together with the liquid-vapour equilibrium.

Several inlet streams may be considered and their temperatures and compositions are given.

SYMBOLS AND CONNECTION RULES



Inlet Streams: Material: maximum number = 8.

Thermal: maximum number = 2

Outlet Streams: Material: 1 stream of state V as vapour outlet (upper stream)

1 stream of state L as liquid outlet (lower stream)

Thermal: maximum number = 2

Thermods: Both outlet streams must use the same Thermod.

Attention:

$Q > 0$ if heat is supplied to the reactor.

$Q < 0$ if heat is removed from the reactor.

UNIT PARAMETERS (SUMMARY)

MODE2	Input	CALCULATION MODE
MODE1	Input	PRESSURE MODE
KPRINT	Input	Print-out Level
ALFOUT	Input	OUTLET VAPOUR FRACTION
TOUT	Input/result	OUTLET TEMPERATURE
POUT	Input/result	OUTLET PRESSURE

UNIT PARAMETERS

MODE2	Input	CALCULATION MODE

If the vessel pressure is calculated according to the procedure described below, three modes are available.

MODE2 = 1	-	FIXED,LOAD The heat provided or removed to the vessel is given. The temperature, flowrates of both liquid and vapour streams are calculated together with both phase compositions.
MODE2 = 2	-	FIXED,TOUT The vessel temperature is given. Flowrates and compositions of both outlets are calculated together with the heat to be removed or provided.
MODE2 = 3	-	FIXED,ALFOUT The molar vapour fraction is given (ratio of vapour and liquid molar flowrates). The temperature and compositions of both phases are calculated together with the heat to be removed or provided.

MODE1	Input	PRESSURE MODE

Several calculation modes are available to calculate the vessel pressure.

MODE1 = 0	-	The vessel pressure is equal to the smallest inlet pressure.
MODE1 = 1	-	The vessel pressure is equal to the smallest inlet pressure multiplied by the expansion factor (POUT) which is given.
MODE1 = 2	-	The vessel pressure is equal to the smallest inlet pressure minus the given pressure drop (POUT).
MODE1 = 3	-	The vessel pressure is equal to the given pressure (POUT).
MODE1 = 4	-	The molar vapour fraction is given (ratio of vapour and liquid molar flowrates) together with the heat to be removed or provided. The temperature, pressure and compositions of both outlets are calculated.

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

ALFOUT	Input	OUTLET VAPOUR FRACTION
TOUT	Input/result	OUTLET TEMPERATURE
POUT	Input/result	OUTLET PRESSURE

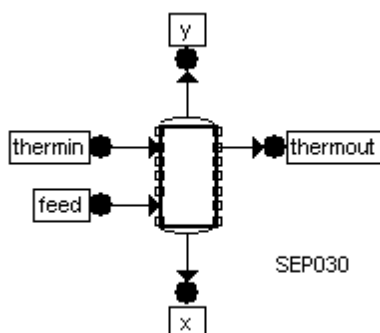
Vessel pressure (MODE2=3) or Pressure drop (MODE2=2) or expansion factor (MODE2=1)

SECTION 39: SEP030 : BLACK BOX SEPARATOR

PURPOSE

The SEP030 UNIT is a 'black box' ensuring the mixing of inlet streams and the separation of this mixture into two outlet streams of different compositions, according to user specified compounds distribution laws. This unit can be used as a short-cut model for any separation equipment.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	From 1 to 10 material streams (any state)
Outlet STREAMS	1	Bottom (X) material stream (Liquid/Vapour)
	2	Top (Y) material stream (Liquid/Vapour)

The two outlet streams must use the same Thermod. In addition, this Thermod must contain all compounds that actually appear in any of the inlet streams.

An additional Thermal stream can be connected either as the last inlet or as the last outlet stream, to close the heat balance.

Attention: By mirroring the symbol vertically, the Y stream will become the (apparent) bottom stream allowing to define the compound split fractions on the (apparent) bottom stream.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation mode
TYPTH1	Input	Y (Top) stream state
TYPTH2	Input	X (Bottom) stream state
KPRINT	Input	Print-out Level
KEY	Input	Mode 3: key compound index in Thermod
AKEY	Input	Mode 3: key compound molar fraction
YTOT	Input	Mode 4: Y (Top) Total molar rate
FRAC _i	Measurable	Split factor for compound <i>i</i>
TOUT _j	Measurable	Temperature difference for stream <i>j</i> (<i>j</i> = 1, 2)
POUT _j	Measurable	Pressure or Pressure difference for stream <i>j</i> (<i>j</i> = 1, 2)
ALFOUT _j	Measurable	Vapour Fraction of stream <i>j</i> (<i>j</i> = 1, 2)

UNIT PARAMETERS

MODE	Input	Calculation mode

This parameter defines the distribution law to be used :

- 1 **$Y_i = \text{FRAC}_i \cdot Z_i$ (*i* = 1,...number of compounds)**

The partial flow rate of each compound in the Y (top) stream is expressed as a fraction of its partial rate in the (total) feed stream.

- 2 **$Y_i / Y_{\text{Tot}} = \text{FRAC}_i \cdot X_i / X_{\text{Tot}}$ (*i* = 1,...number of compounds)**

The molar fraction of each compound in the Y (top) stream is expressed in function of its molar fraction in the X (bottom) stream.

- 3 **$Y_i = \text{FRAC}_i \cdot Y_{\text{Tot}}$ and $X_{\text{key}} = \text{AKEY} \cdot X_{\text{Tot}}$ (*i* = 1,...number of compounds)**

The parameters FRAC_i express the composition of the Y (top) stream. In addition the molar fraction of a key compound is given for the X (bottom) stream.

- 4 **$Y_i = \text{FRAC}_i \cdot Y_{\text{Tot}}$ and Y_{tot} given** (*i* = 1,...number of compounds)

The parameters FRAC_i express the composition of the Y (top) stream. In addition the total molar flow rate of the top stream is given in the parameter YTOT.

TYPTH1	Input	Y (Top) stream state
TYPTH2	Input	X (Bottom) stream state

This parameter defines how the state of the corresponding outlet stream is calculated

0 Mass balance only

The stream intensive values (T,P...) are unchanged

1 Given pressure and temperature

The stream is set and the pressure and temperature given in the parameters POUT_{*i*} and TOUT_{*i*}.

For a single compound mixture the parameter ALFOUT_{*i*} will also be used.

2 Given pressure and vapour fraction

The stream is set at the pressure and vapour fraction given in the parameters POUT_{*i*} and ALFOUT_{*i*}.

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

KEY	Input	Mode 3: key compound index in Thermod

AKEY	Input	Mode 3: key compound molar fraction

In Mode 3, the parameter AKEY indicates the molar fraction of KEYcompound in the X (bottom) stream.

YTOT	Input	Mode 4: Y (Top) Total molar rate

In Mode 4, the parameter YTOT gives the total molar flow rate of the Y (top) stream.

FRAC_{<i>i</i>}	Measurable	Split factor for compound <i>i</i>

For each compound (in the order set in the Thermod) the factor FRAC_{*i*} must be specified.

TOUT_{<i>j</i>}	Measurable	Temperature difference for stream <i>j</i> (<i>j</i> = 1, 2)

POUT_{<i>j</i>}	Measurable	Pressure or Pressure difference for stream <i>j</i> (<i>j</i> = 1, 2)

ALFOUT_{<i>j</i>}	Measurable	Vapour Fraction of stream <i>j</i> (<i>j</i> = 1, 2)

These parameters are used to set the state of the outlet streams as defined by the parameter TYPTH_{*j*}.

If Pout > 0 set pressure

If Pout = 0 outlet pressure = minimum of inlet pressure

If Pout < 0 outlet pressure = minimum of inlet pressure + Pout

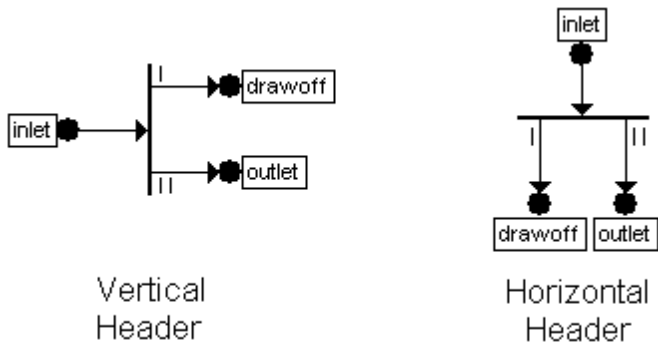
Attention: when POUT is used as a pressure drop ($\text{POUT} < 0$), it must be expressed as an absolute pressure. Specifying POUT in barg or psig would produce wrong results.

SECTION 40: SEP050 : SPLITTER

PURPOSE

The SEP050 unit allows you to draw-off, from a given stream, a secondary stream with the same composition. This drawn-off stream is the first outlet stream.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1	Material stream (any state)
Outlet STREAMS	1	Draw-off material stream
	2	Second outlet material stream

All the streams should use the same Thermod and state.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation mode
KPRINT	Input	Print-out Level
FLOWRATE	Measurable	Draw-off flowrate (first outlet)

UNIT PARAMETERS

MODE	Input	Calculation mode

0 Heat and mass balance

The outlet streams are set in the same state as the inlet stream

1 Mass balance only

The state of the outlet streams is left unchanged.

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

FLOWRATE	Measurable	Draw-off flowrate (first outlet)

FLOWRATE > 0	Set-point on the flow rate to be draw off.

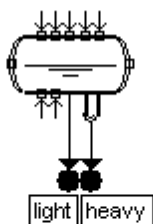
FLOWRATE = -1	The set-point on the flow-rate to be draw off is taken from the current flow rate of the first outlet stream itself.

SECTION 41: SEP060 : LIQUID/LIQUID SEPARATOR

PURPOSE

The unit **SEP060** is a separation drum for a liquid/liquid mixture. Entrainment between the 2 phases can be modelled. Different calculation modes are available (adiabatic or given operating temperature).

SYMBOLS AND CONNECTION RULES



Optional Connections:

- *Thermal Stream*
- ⏏ *Material Stream*

Inlet STREAMS	1-10	Inlet material streams. (LL or LLV state, although other states can be used for streams that do not involve two liquid phases).
Outlet STREAMS	1	Heavy liquid material stream (LL or LLV state, L is also accepted if there is no entrainment between the two phases).
	2	Light liquid material stream (LL or LLV state, L is also accepted if there is no entrainment between the two phases).
	3	Thermal stream (refused in Mode 1, mandatory otherwise)

All streams must use the same Thermod.

Attention:

- Should the phase equilibrium calculation result in a single liquid phase, the flow rate of the light liquid stream (second outlet) is set to zero.
- If a vapour phase appears, an error message is issued and the simulation is stopped.

UNIT PARAMETERS (SUMMARY)

MODE	Input	Calculation mode
KPRINT	Input	Print-out Level
TOUT	Measurable	Operating temperature
HEAVYRAT	Measurable	Heavy to light phase entrainment rate
LIGHTRAT	Measurable	Light to heavy phase entrainment rate

UNIT PARAMETERS

MODE	Input	Calculation mode

MODE = 1 Adiabatic separator.

MODE = 2 The quantity of heat to be transferred is supplied.

MODE = 3 Isothermal separator. The quantity of heat to be supplied or to be draw-off is computed.

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

TOUT	Measurable	Operating temperature

Used only when MODE=3

HEAVYRAT	Measurable	Heavy to light phase entrainment rate

This parameter ranges from 0 up to 1 and indicates the fraction of the heavy liquid phase that is entrained in the light liquid phase. This allows to model non-ideal separators.

LIGHTRAT	Measurable	Light to heavy phase entrainment rate

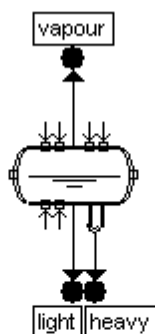
This parameter ranges from 0 up to 1 and indicates the fraction of the light liquid phase that is entrained in the heavy liquid phase. This allows to model non-ideal separators.

SECTION 42: SEP070 : THREE PHASE SEPARATOR (L/L/V)

PURPOSE

The unit **SEP070** is a liquid/liquid/vapour separation drum.. Entrainment between the phases can be modelled. Different calculation modes are available (set pressure or vapour fraction).

SYMBOLS AND CONNECTION RULES



Optional Connections:

- *Thermal Stream*
- ⬇ *Material Stream*

Inlet STREAMS	1 to 10 inlet streams of any standard format (LV, LLV, WL, V, THERSTRM,...).
Outlet STREAMS	<i>Outlet Vapour</i> STREAM. (3) Format V if there is no entrainment of the liquid phases. <i>Light Liquid</i> STREAM. (2) Format LL or WL. L is also accepted if there is no entrainment of the vapour phase or of the other liquid phase respectively. <i>Heavy Liquid</i> STREAM. (1) Format LL or WL. L is also accepted if there is no entrainment of the vapour phase or of the other liquid phase.

Attention:

1. All components present in the inlet Streams should also be found in the outlet Streams.
2. Should the phase equilibrium calculation result in foretelling only one liquid phase, calculations are resumed by setting the flowrate of the second outlet to 0.

UNIT PARAMETERS

MODE	Input	Pressure mode
MODE = 0	-	If there is no pressure drop.
MODE = 1	-	PCHANGE is an expansion factor.
MODE = 2	-	If PCHANGE is a pressure drop (DP).
MODE = 3	-	If the tank pressure is set to PCHANGE.
MODE = 4	-	If you want a vapour fraction VAP.FRAC to be set (Then the related outlet pressure is calculated).

PCHANGE	Input	Pressure
if MODE = 1	-	Expansion factor (applied to the lowest inlet P).
if MODE = 2	-	Pressure drop (applied to the lowest inlet P).
if MODE = 3	-	Set pressure.

ITMAX	Input	MAX NBR OF ITERATIONS

Maximum number of iterations for pressure computation.

20 is the default value.

KPRINT	Input	Print-out Level

Controls the amount of information printed in the trace file.

HEAVYRAT	Input	ENTRAINMENT RATE OF HEAVY PHASE INTO LIGHT PHASE

This parameter range is from 0 up to 1 (0\H\1) If HEAVYRAT < 0, default value 0 is chosen by program.

Note : The calculation of equilibrium composition and outlet flow rates is based on the assumption that separation between liquid phases is perfect. Then a branch fraction equal to the entrainment rate is added to each outlet branch.

LIGHTRAT **Input** **ENTRAINMENT RATE OF LIGHT PHASE INTO HEAVY PHASE**

If $LIGHTRAT < 0$, then the default value of 0 is chosen.

VAPRAT **Input** **ENTRAINMENT RATE OF VAPOUR INTO LIQUIDS**

If < 0 , then the default value of 0 is chosen.

LIQRAT **Input** **ENTRAINMENT RATE OF LIQUIDS INTO VAPOUR**

If < 0 , then the default value of 0 is chosen.

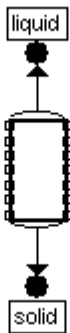
Note : Entrainment rates are applied to the liquids first, then to the vapour phase.

SECTION 43: SEP080 : SOLID/FLUID PHASE SEPARATOR

PURPOSE

UNIT SEP080 is a model for a device separating solid phases from fluid phases. Several feed streams are possible.

SYMBOLS AND CONNECTION RULES



Inlet STREAMS	1-10	Material streams (any state)
Outlet STREAMS	1	Material stream for the solids (bottom stream)
	2	Material stream for the fluid (top stream). Its state should be set in accordance with the inlet stream.

The two outlet streams must use the same Thermod.

All compounds defined in any of the Thermod of the inlet streams should be defined as well in the Thermod of the outlet streams.

Attention: by mirroring the symbol vertically, the solid stream will be at the top and the fluid stream will be at the bottom.

UNIT PARAMETERS (SUMMARY)

KPRINT	Input	Print-out Level
LIQRATE	Input	Entrainment rate of the fluid phase into the solid phase
SOLRATE	Input	Entrainment rate of the solid phase into the fluid phase

UNIT PARAMETERS

KPRINT **Input** **Print-out Level**

Controls the amount of information printed in the trace file.

LIQRATE **Input** **Entrainment rate of the fluid phase into the solid phase**

Fraction of fluid compounds that is trapped into the solid stream, expressed as a fraction of the total inlet flow rate of the fluid compounds.

If LIQRATE < 0, then the default value of 0 is used.

SOLRATE **Input** **Entrainment rate of the solid phase into the fluid phase**

Fraction of solids compounds that is entrained into the fluid stream, expressed as a fraction of the total inlet flow rate of the solid compounds.

If SOLRATE < 0, then the default value of 0 is used.

SECTION 44: SIMFLX : USER-DEFINED CALCULATIONS

PURPOSE

The **SIMFLX** unit allows :

- adding user defined calculations
- adding user defined processing like reporting, printing,
- defining up to ten parameters, used by the same unit or by any other unit in the same model

Using **FLEX** language, you can access values of streams and units and perform calculations. Thermodynamic routines can be used to calculate various thermodynamic properties and perform physical equilibrium compositions.

A **FLEX** program can also be added to most of the **SIMU** units. It is possible to limit to execute the code before, after or both before and after the code the unit itself is executed.

Tags can be defined on the parameters of the **SIMFLX** unit, making it possible to use them as either controlled or manipulated variables of various controllers.

SYMBOLS AND CONNECTION RULES



No stream can be connected to a unit of type **SIMFLX**.

UNIT PARAMETERS

VARi **Measurable** **Multi-purpose parameters**

These parameters can be used to store various calculation results or user settings.

44.1 DEFINING THE FLEX PROGRAM

CREATING A FLEX PROGRAM

- Define the unit of type **SIMFLX**.

- Write your FLEX program in a separate file.
- When starting to calculate your model, enter the name of the file into the corresponding field

Attention: When SIMU saves the results after a successful run, the Flex program is written into the PDB file. The program is then kept for further runs and it is no longer necessary to reload the Flex program. To modify the Flex program, you must reload it. It is also possible to add a Flex code to most of SIMU units.

STRUCTURE OF A FLEX PROGRAM

The Flex program attached to a Unit consists typically of:

a NOCALL string:	defines when the Flex code should be executed
a declaration section:	defines Variables and Arrays that will be necessary
selection of a Thermod:	defines the thermodynamics methods to be used.
selection of variables:	defines which variables will be used in the equation.
calculations:	defines the calculations.

Comments can be entered anywhere after the exclamation mark "!".

Flex syntax is similar to FORTRAN Code. Chapter 6 describes this syntax in detail.

Example: &&&COMMENT FLX1 ! File Header for flex code of unit FLX1

NOCALL = ...

! Declaration

REAL ...

! Selection of thermodynamic methods

CALL CHOOSE('Astream')

! Selection of variables

CALL LOCATE1(...)

CALL LOCATEN(...)

! Calculations

...

END

...

&&&END ! end of flex code of unit FLX1

...

&&&EOF ! end of file

44.2 DEFINING WHEN THE CODE MUST BE EXECUTED

By default the Flex code is processed:

- in verification phase (Phase 1),
- before and after each calculation of the unit in simulation phase (Phase 3),
- before and after calculation of the unit in the report phase (Phase 4).

The NOCALL variable is used to suppress some of these calls, by allocating it a string of up to eight characters, obtained by concatenating the following strings:

IS	suppress the code execution before unit calculation in simulation phase
OS	suppress the code execution after unit calculation in simulation phase
IR	suppress the code execution before unit calculation in report phase
OR	suppress the code execution after unit calculation in report phase

Example: to suppress the flex code execution before unit calculation used the following instruction:

```
NOCALL = 'ISIR'
```

44.3 FLEX DECLARATIONS OF VARIABLES

PREDEFINED VARIABLES

IHELP	error flag: the calculation stops if this variable is set to a non zero value										
PHASE	indicates the current phase: <table style="margin-left: 20px;"> <tr> <td>1</td><td>verification,</td></tr> <tr> <td>3</td><td>simulation (before unit calculation)</td></tr> <tr> <td>-3</td><td>simulation (after unit calculation)</td></tr> <tr> <td>4</td><td>simulation (before unit calculation)</td></tr> <tr> <td>-4</td><td>simulation (after unit calculation)</td></tr> </table>	1	verification,	3	simulation (before unit calculation)	-3	simulation (after unit calculation)	4	simulation (before unit calculation)	-4	simulation (after unit calculation)
1	verification,										
3	simulation (before unit calculation)										
-3	simulation (after unit calculation)										
4	simulation (before unit calculation)										
-4	simulation (after unit calculation)										
NIN	number of inlet streams of the unit										
NOUT	number of outlet streams of the unit										
UNINAM	name of the unit										
NPAR	number of parameters of the unit										
UNIPAR	array with unit parameters values										
UNIPARNM	array with unit parameters names										

NAME XXX(N), YYY

declares characters*8 arrays and scalars.

xxx can be any name, n will be the size of array xxx,

yyy can be any name; yyy will be a single CHARACTER*8.

Example: NAME NAMES(3), ANAME

REAL XXX(N), YYY

declares REAL arrays and scalars.

xxx can be any name, n will be the size of array xxx,

yyy can be any name; yyy will be a scalar.

Example: REAL VAP(3), LIQ(3), T, P

44.4 FLEX ROUTINE TO SELECT THERMODYNAMIC MODEL**CALL CHOOSE('NAME') OR CALL CHOOSE(N)**

This subroutine selects a Reference stream defining which methods to use for thermodynamic functions. A Reference stream must be defined before functions, like EQPHYS and PROPTY, can be used. The stream can be selected by its name or by a number indicating its position in the inlet streams (positive number) or in the outlet streams (negative number).

The following FLEX variables and arrays are initialised when using the CHOOSE function:

Name	STRNAM	Inlet STREAM NAME.
Name	STTYPE	Inlet STREAM TYPE.
Name(i)	STRPARNM	Inlet STREAM PARAmeters NaMes (array).
Name	RATESNAM	Associated RATES NAME.
Name	RATESTYPE	Associated RATES TYPE (i.e. : MOL or MASS).
Name(i)	RATPARNM	Associated RATES NaMes of the components (array).
Real	NUMSUB	NUMber of components (SUBstances).
Name(i)	SUBTYPE	Components (SUBstances) TYPE (i.e. : EPIC or SOLID).
Real(i)	STRPAR	Array with stream parameters
Real(I)	RATESPAR	Array with rates parameters.

For non material streams, the STRPAR array contains the variables of the stream and the RATESPAR is undefined.

For material streams, STRPAR and RATESPAR have the following structure:

STRPAR ARRAY :

(1)	T	Temperature value (K).
(2)	P	Pressure value (bar).
(3)	VAPF	VAPour Fraction value (not used).
(4)	RATEFRAC	Rates fraction (MIXF) value.
(5)	H	Enthalpy value (kJ/kmol).
(6)	S	Entropy value (not used).

Attention: The structure of STRPAR allows you to use it directly to call EQPHYS.

RATESPAR ARRAY :

Contains the total and partial flow rates associated to the Stream in the following form.

RATESPAR(1) :	total molar flow rate (kmol/s)
RATESPAR(2) :	total mass flow rate (kg/s)
RATESPAR(3,4,...)	partial molar flow rate of component 1,2,... (kmol/s)

Attention:

1. To obtain the real flow rates of the stream, you must multiply the elements of RATESPAR by the Rates fraction (RATEFRAC).
2. The array may be used as an argument for EQPHYS and PROPTY by giving RATESPAR(3) as an argument.

Example : `CHOOSE('AFLX-3')` ! loads stream AFLX-3 as the reference stream.
 `CHOOSE(1)` ! loads the first inlet stream as the reference stream.

44.5 FLEX SUBROUTINES TO SELECT VARIABLES

INTRODUCTION

The procedure LOCATE1 and LOCATEN create a link between a local variable (or array) of the Flex code and a variable (or array) corresponding to a unit, stream or compound. After the link is created, the local Flex variable and the corresponding object variable are pointing to the same physical memory location. This means that changing the Flex variable value will change the corresponding object variable and vice-versa.

The procedure ENTRY returns the name of an object from its pointer in the dictionary.

CALL LOCATE1(MYVAR,'INFO','TYPE','OBJECT' OR N,'PAR' OR I)

This function links the local variable MYVAR to the variable defined by 'INFO', 'TYPE', either 'OBJECT' or N, and either 'PAR' or I.

'INFO' can be:

- 'S' : structure record
- 'D' : data record (parameters)

'TYPE' can be:

- 'U' or 'UNIT' : the object is a unit
- 'S' or 'STREAM' : the object is a stream
- 'R' or 'RATES' : the object is the Rates record associated to a stream
- 'C' or: the object is a compound
- 'COMPOUND'

'OBJECT' is the object name

N for a stream, indicates his position as inlet (positive) or outlet (negative) stream.

'PAR' is the name of the variable (or parameter)

I is the position of the variable (or parameter) in the record

Examples CALL LOCATE1(Temp,'D','S','F1','T')

! The local variable Temp is now linked to the temperature of stream F1

CALL LOCATE1(Pres,'D','S',1,2)

! The local variable Pres is now linked to the second variable of the first inlet stream

! (its pressure if a material stream)

CALL LOCATEN(MYARRAY,'INFO','TYPE','OBJECT' OR N)

This function links the local array MYARRAY to the variable defined by 'INFO', 'TYPE' and either 'OBJECT' or N.

'INFO' can be:

- 'S' : structure record
- 'D' : data record (parameters)

'TYPE' can be:

- 'U' or 'UNIT' : the object is a unit
- 'S' or 'STREAM' : the object is a stream
- 'R' or 'RATES' : the object is the Rates record associated to a stream
- 'C' or: the object is a compound
- 'COMPOUND'

'OBJECT' is the object name

N for a stream, indicates his position as inlet (positive) or outlet (negative) stream.

Examples CALL LOCATEN(STREAM,'D','S','F1')

! The local array STREAM is now linked to the data record of stream F1

CALL LOCATEN(STREAM,'D','S',-2)

! The local array STREAM is now linked to the data record of the second outlet stream

CALL ENTRY(NUM,NOM)

The procedure returns, in the variable NOM (of NAME type), the name of the object stored in location NUM of the dictionary of the PDB.

CALL GTTAGVAL('NAME','VALUE',MYVAR)

This function copies the requested value associated with the tag into the local variable MYVAR. These values do not change between iterations.

'NAME' is the Name of the TAG from which the value is to be obtained.

'VALUE': reserved keyword with the name of the value of the TAG to access.

MEA.VAL	the measured value
MEA.ACC	the measured accuracy
SL	the solver lower bound
LL	the alarm lower bound
L	the warning lower bound
H	the warning upper bound
LL	the alarm upper bound
SL	the solver upper bound

MYVAR : Variable to store the result.

Example: CALL GTTAGVAL('TI-001','MEA.VAL',TI_MEAS)
 'TI-001' is the TAG Name,
 'MEA.VAL' selects the measured value
 TI_MEAS is the result variable.

44.6 FLEX SUBROUTINES FOR THERMODYNAMIC CALCULATIONS

CALL EQPHYS(VEC1, VEC2, VAP, LIQ, SOL, LIQ2, 'TYPE')

This function allows to compute phase equilibrium. It is necessary to select a reference stream (with the CHOOSE function) previously.

VEC1	array with stream variables.
VEC2	array with partial molar flow rates of each component.
VAP	array including after computation the partial molar flow rates of the vapour phase.
LIQ	array including after computation the partial molar flow rates of the (first) liquid phase.
SOL	array including after computation the partial molar flow rates of the solid phase.
LIQ2	array including after computation the partial molar flow rates of the (second) liquid phase.
'TYPE'	a reserved KEYWORD which defines the type of equilibria (flash) to perform.

DETAILED DESCRIPTION

VEC1 array with stream variables.

VEC1(1)	temperature (K).
VEC1(2)	pressure (bar).
VEC1(3)	vapour fraction ($VAPF = (V/V+L)$).
VEC1(4)	not used.
VEC1(5)	enthalpy flow rate (kW).
VEC1(6)	entropy flow rate (kW/K).

Attention:

1. It is mandatory to give initial values to variables used with the flash type defined, others can be given as initial values.

2. The STRPAR array created by the GETALL (or GETVAR) subroutine has the same structure and may be used as the first argument for EQPHYS.

VEC2: Array with partial molar flow rates of each component (kmol/s). This array may be initialised with a call to GETALL or to GETVAR.

VAP: Array with a size of at least the number of components of the structure, including after computation the partial molar flow rates of the vapour phase (kmol/s).

LIQ: Array with a size of at least the number of components of the structure, including after computation the partial molar flow rates of the (first) liquid phase (kmol/s).

SOL: Array with a size of at least the number of components of the structure, including after computation the partial molar flow rates of the solid phase (kmol/s).

LIQ2: Array with a size of at least the number of components of the structure, including after computation the partial molar flow rates of the (second) liquid phase (kmol/s).

'TYPE': A KEYWORD which defines the type of equilibrium (flash) to perform:

EPT:	equilibrium at a given P and T.
EPA:	equilibrium at a given P and vapour fraction.
ETA:	equilibrium at a given T and vapour fraction.
EHP:	equilibrium at a given P and enthalpy.
ESP:	equilibrium at a given P and entropy.
EPTA:	Extension of EPT with vapour ratio given as an argument but only used in the saturation zone of a single component stream, where T and P do not uniquely define the stream state.

The number of phases to be taken into account for the equilibrium computation is set with the variable STTYPE, initialised by CHOOSE from the type defined for the stream in the PDB.

It is possible to change it to compute other flash types: V, L, S, LV, LL, LLV, WL, WLW.

CALL PROPTY('FUNCTION', T, P, VEC3, MYVAR)

This subroutine allows to compute physical and thermodynamic properties. It is necessary to previously select a reference stream (with the CHOOSE function).

'FUNCTION' : A KEYWORD which defines the requested computation.

TOTMAS	:	total mass flow rate (kg/s).
TOTMOL	:	total molar flow rate (kmol/s).
MOLFRA	:	molar ratios (mol/mol).
MASFRA	:	mass ratios (kg/kg).
MOLDRY	:	molar ratios of dry gases.
MASFLW	:	partial mass flow rates (kg/s).

For vapour phase properties :

VG	:	volumetric flow rate (m ³ /s).
HG	:	enthalpy flow rate (kW).
GG	:	free energy flow rate (kW).
SG	:	entropy flow rate (kW/K).
CPG	:	thermal capacity of stream (kW/K).
CONG	:	thermal conductivity (kg/m/s).
VISG	:	dynamic viscosity (kg/m/s).
SONG	:	speed of sound (m/s).
FUGG	:	fugacity of components (bar).

For liquid phase properties :

VL	:	volumetric flow rate (m ³ /s).
HL	:	enthalpy flow rate (kW).
GL	:	free energy flow rate (kW).
SL	:	entropy flow rate (kW/K).
CPL	:	thermal capacity of stream (kW/K).
CONL	:	thermal conductivity (kg/m/s).
VISL	:	dynamic viscosity (kg/m/s).
SONL	:	speed of sound (m/s).
FUGL	:	fugacity of components (bar).

For solid phase properties :

VS	:	volumetric flow rate (m ³ /s).
HS	:	enthalpy flow rate (kW).
GS	:	free energy flow rate (kW).
SS	:	entropy flow rate (kW/K).
CPS	:	thermal capacity of stream (kW/K).
CONS	:	thermal conductivity (kg/m/s).
FUGS	:	fugacity of components (bar).

T: Temperature (K).

P: Pressure (bar).

VEC3: Array with partial molar flow rates of components (kmol/s).

MYVAR: Variable to store the MYVAR. It is an array with a size of at least the number of components when requesting for function of composition or fugacity.

CHAPTER 9

VALIREPORT

SECTION 1: INTRODUCTION

1.1 WHAT CAN YOU DO WITH THIS APPLICATION?

ValiReport allows you:

- to store in a database (ACCESS or MS-SQL) the values of a number of Tags for successive runs you perform with VALI: measurement, measured precision, validated value, validated precision, filtered value, filtered precision, penalty, gain, status (filtered, eliminated, etc.) physical unit and comment
- to retrieve all the values stored in the database for a given run
- to retrieve aggregated values of all the run between two dates
- to retrieve selected tag values for a number of runs
- to produce charts showing the evolution of selected Tag values with time
- to design or import reports and flow-sheets of a process and display selected Tags values or aggregated values
- to colour displayed measurement of a flow-sheet if it is suspicious i.e. with high Penalty
- to run and interact with ValiEngine

The application uses Excel Workbooks and a number of procedures stored in an Excel Add-In.

ValiReport is interfaced with ValiEngine, which provides the tags values in a run data file named from the name of the main PFD (e.g. main.arc) and with ValiModeller, which writes the structure file (e.g. main.str) which relates each tag to one or two model units.

In an online environment, the tags values will be automatically loaded in the database using the ValiDBLoader application, while in an offline environment, you will directly load the data in your database from ValiReport itself.

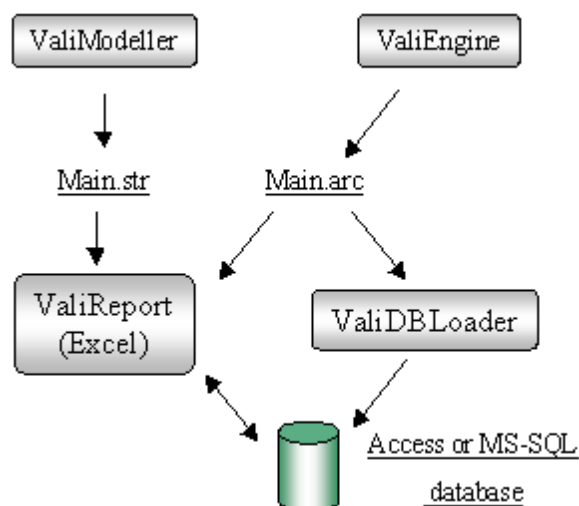


FIGURE 1 - VALIREPORT ENVIRONMENT

1.2 REQUIREMENTS

For a proper use of ValiReport, you should satisfy the following requirements:

Software requirements:

Microsoft Windows NT4 (SP6a), 2000, XP, 2003, Vista

Microsoft Excel 97 or higher version (including 2007)

For Access database: Microsoft Access 97, 2000, 2002 (part of Office XP) or 2003/2007 (a runtime is sufficient).

For SQL-Server: Microsoft SQL Server 2000 SP4 or Higher.

Before you can use ValiReport Add-in, you must install it on your computer (see installation procedure) and then load it in Microsoft Excel. To load ValiReport use the **Tools|Add-ins** Menu command and select ValiReport add-in. If it is not listed in the **Add-Ins available** box, click Browse, and select it in the ValiReport installation folder. Once loaded, the command **ValiReport** will appear in the Tools menu.

1.3 SPECIAL FEATURES AND CONVENTIONS

This Manual uses the following style for Excel menu commands: **File|Open...** . This means that you pull down the **File** menu and select the **Open...** command. Depending on the installation language of your Office applications, some menu commands might appear in a different language. For example in a French installation, the menu command **Tools** will be displayed as **Outils**.

Some menus and menu items have accelerator keys that appear underlined in the menu bar.

The most important tasks can be executed by using either the menu bar or the toolbar.

SECTION 2: HANDLING VALIREPORT WORKBOOKS

2.1 CREATING A VALIREPORT WORKBOOK

You can create as many databases and ValiReport Workbooks as you want. Several Workbooks can share the same database. However, a given ValiReport Workbook can only access one database at a time.

Should you use SQL Server to store your data, you need to perform some configuration work before creating any ValiReport as described in sub-section "SQL-Server database".

You can create a ValiReport Workbook from any Excel Workbook, using the following procedure:

- select the Tools|ValiReport menu command
- in the Save As dialog box, enter the name you wish to give to your ValiReport workbook and click Save

By default, ValiReport uses the following sub-folders (starting from the folder where you saved the workbook):

- .\Access: folder where the ACCESS databases are stored
- .\Archive: unused at this time

The arc files are assumed to be located in the same folder as the ValiReport excel file. If you want to use other folders, you will need to change these settings afterwards.

- specify the name or IP address of the Belsim LicenseServer and the type of database (Access or SQL Server) you will use. Click **OK** to proceed

Attention: The default address of the license server is read from the Belsim.ini file in the dat folder of the Vali installation

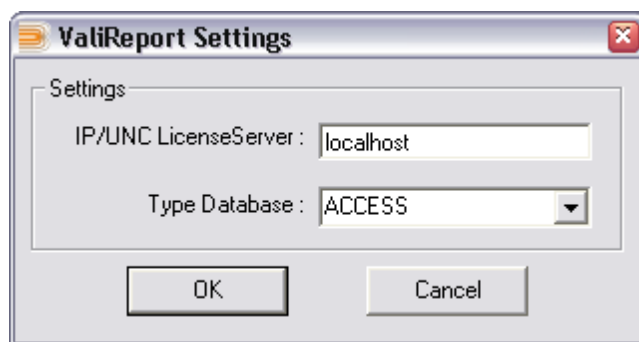


FIGURE 2 - VALIREPORT CREATION SETTINGS

- if you select a SQL Server database, you will be prompted to define the plant name as defined in Belsim License Manager (e.g. SQLPLANT) and the database name as defined in the SQL Server (e.g. validata). Click **OK** to proceed

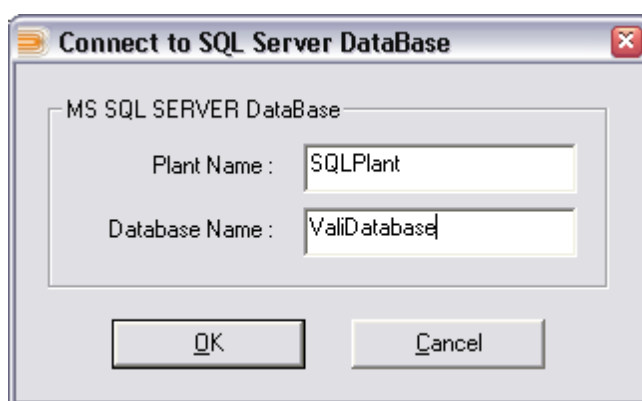


FIGURE 3 - VALIREPORT DATABASE CONFIGURATION

- if you selected an ACCESS database and if no database exists in the Access folder, you will be prompted to give a name for the new database.

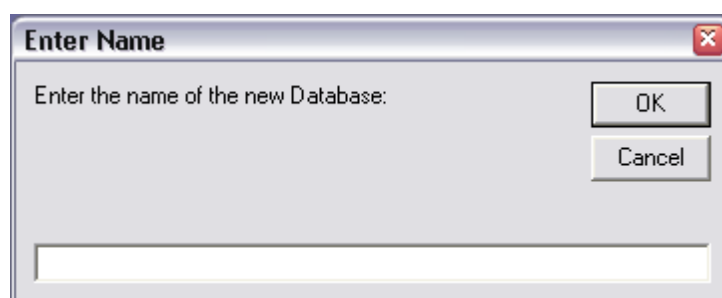


FIGURE 4 - CREATING A NEW ACCESS DATABASE

- if a database already exists in the Access folder, you will be prompted to select an existing database.

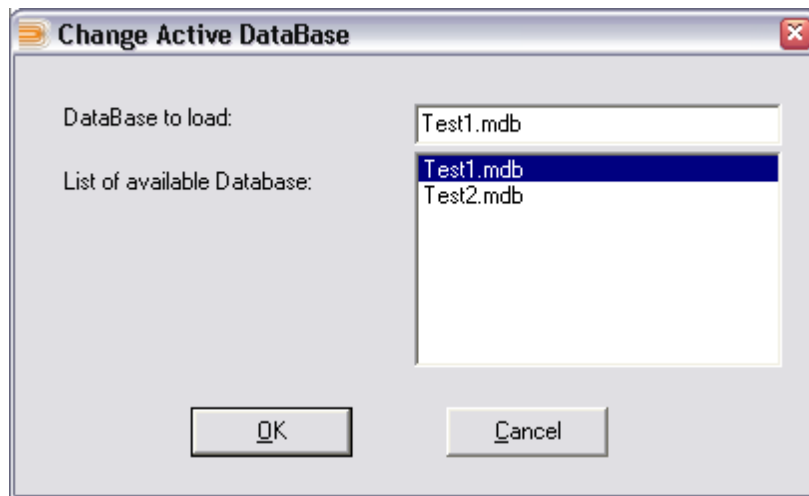


FIGURE 5 - SELECTING AMONG EXISTING DATABASES

- select a database from the list and click **OK**
- or
- enter a new name to create a new Access database and click **OK**

You are then ready to proceed with your ValiReport application. The next time you will open the corresponding Excel file, it will be recognised as a ValiReport application. The configuration information is stored in an ini file with the same name as the ValiReport workbook.

2.2 OVERVIEW OF VALIREPORT

ADDITIONAL MENU ITEMS AND TOOLBAR

The menu bar presents the following items added to the Excel menu bar features:

Database, Runs, Application, ValiReport Help

The most useful commands are also grouped on the **ValiReport** toolbar:

- Database:
 - Change Active DB...: change the active database
 - Rename active DB...: rename the active database
 - Create DB...: create a new database
 - Delete DB...: delete a database
 - Compact Active DB: compact the active database (Access only)
- Runs:
 -  Load Run...: import a run data file in ValiReport
 -  Insert Run...: insert a new Run in the database
 - Auto Insert Run: insert a new Run in the database
 -  Retrieve Run...: retrieve a Run from the database
 -  Create Table and Chart: create Tables and Charts
 - Remove Run...: remove Runs from the database
- Application:
 - Edit VALI Tag...: launch the EDIT tag window
 -  Run Vali...: launch ValiOfflineEngine
 - Options...: customize your ValiReport application
 - Unmark ValiReport: convert the ValiReport workbook into a standard excel workbook
 - Display |ON/OFF Status: display object status in RUN sheet
 - Display |Struct Sheet: create a structure sheet showing the tags sorted out by units
 - Options...: customize your ValiReport application
- ValiReport Help:
 - Contents and Index: activate the Help file
 - Belsim Home page: access the Belsim Home Page
 -  About Belsim S.A.: display Belsim coordinates

THE SHEETS RUN AND STRUCT

In the RUN sheet, you can load the data of a single Vali run before storing it into the database. You also use this sheet to retrieve a complete run from Vali Database.

By default, when opening a ValiReport, the last run (on basis of its date and time) will be loaded from the Vali Database.

Optionally, a sheet called STRUCT can be created and presents the tags grouped by units.

See section [Working Run by Run](#) for more information.

REPORTS AND FLOW-SHEETS

The application enables to develop customized reports with data coming either from a single run or aggregated between two selected dates. These tag references can be directly selected within your worksheet through a dialogue box that you can activate by a double-clicking the cell where you want to insert the tag reference. Optionally, ValiReport can flag in a different colour the cells referring to suspect tags.

For more information see section [Flow-Sheets and Reports](#).

TABLES AND CHARTS

The **Create Table and Chart** dialog box enables creating Tables and Charts with the data of several tags over several runs. These table and charts are created in individual worksheets of your workbook.

See section [Tables and Charts](#) for more information.

2.3 MOVING AND COPYING A VALIREPORT WORKBOOK

You can copy (or move) your ValiReport workbook as you wish, provided you copy (or move) its associated **ini** file at the same time.

Attention: In a given Excel session, you can use only one ValiReport workbook. To work with several ValiReport workbooks, you need to start several Excel sessions.

2.4 WORKING IN READ-ONLY MODE

Setting the parameter **ReadOnly** of the ini file to 1, will set the ValiReport workbook in Read-only mode, deactivating all menu commands related to ValiDatabase as well as the menu commands enabling to store or delete runs from ValiDatabase.

SECTION 3: HANDLING DATABASES

The operations you can perform from ValiReport depends on the type of database you use: Access or MS-SQL.

3.1 ACCESS DATABASE

CREATING A DATABASE

You create a new database by choosing the menu item **Database|Create DB...** . Within the displayed dialogue box you must enter the name for the new database.

It will initialise the new database with the necessary tables and relationships.

CHANGING THE ACTIVE DATABASE

To change a workbook's active database, execute menu command **Database|Change Active DB...** .

DELETING A DATABASE

The same way you create a database you can delete an existing one, by giving or by choosing the database name in the dialog box. Therefore choose **Database|Delete DB...** from the menu bar and the same dialogue box, as for creating a database, will appear.

Take care that all data stored in the Database will be lost forever, except if you have a backup.

CHANGING THE NAME OF ACTIVE DATABASE

To rename the active database, execute menu command **Database|Rename Active DB** from the menu bar, and give the new name.

COMPACTING THE ACTIVE DATABASE

To compact the active database, execute menu command **Database|Compact Active DB** from the menu bar.

If you delete runs from a database, it is recommended to compact it in order to decrease its size.

3.2 SQL-SERVER DATABASE

PRELIMINARY CONFIGURATION

If you wish to use a MS-SQL server database, you will need to create the SQL database using ValiDatabaseCreator application (accessible from Vali 4 Windows menu).

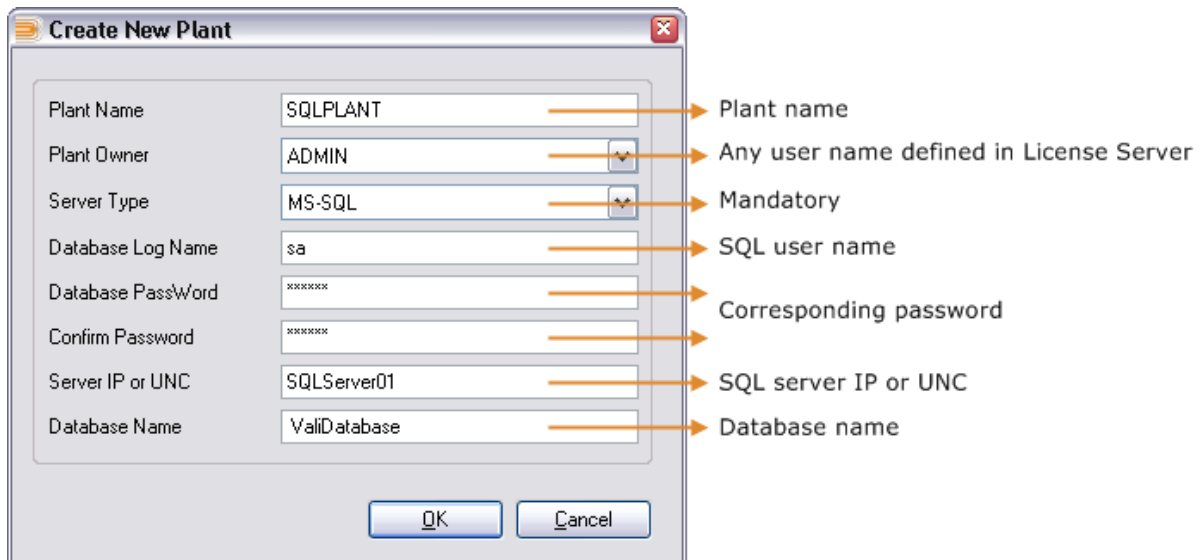
FIGURE 6 - CREATING A SQL SERVER VALIDATABASE

- SQL server authentication (mandatory):
 - IP/UNC: IP/UNC of the SQL server
 - User name: login name of administrator account used to create the database
 - Password: password of the administrator account
- ValiDataBase (mandatory):
 - Create DataBase & Tables: a new Database must be created at the same time
 - Create Tables: only the tables need to be created
 - DB Name: name of the SQL database
 - ValiProduction : tables needed for ValiProduction have to be created
- Create Plant (optional): must be checked if you want to create a new plant in Belsim license server in order to access the database. If not checked, it is assumed you will use a plant already defined in Belsim license server:
 - LicenseServer : IP / UNC of Belsim LicenseServer
 - Plant Name: name of the plant to be created
- Create SQL User check-box: if checked the (Login) user will be created in the SQL server:
 - Login: user name to be used to connect to the SQL database
 - Password: corresponding password

- Confirm password: password confirmation

The **Show SQL Script** button shows the SQL script code that needs to be executed. This enables to execute the corresponding script at the SQL server management tool level.

The Plant can also be defined through Belsim LicenseManager under the **Plant** tab:



The screenshot shows a 'Create New Plant' dialog box with the following fields and annotations:

Field	Value	Annotation
Plant Name	SQLPLANT	Plant name
Plant Owner	ADMIN	Any user name defined in License Server
Server Type	MS-SQL	Mandatory
Database Log Name	sa	SQL user name
Database PassWord	xxxxxx	Corresponding password
Confirm Password	xxxxxx	
Server IP or UNC	SQLServer01	SQL server IP or UNC
Database Name	ValiDatabase	Database name

Buttons: OK, Cancel

FIGURE 7 - CONFIGURING AN SQL SERVER PLANT

CHANGING THE ACTIVE DATABASE

To change a workbook's active database, execute menu command **Database|Change Active DB...** and enter the new database name.

3.3 DATABASE STRUCTURE

The ValiReport database contains 5 tables:

- Runs: List of runs with their Date, and the associated model file, problem (main PFD), model and Vali Engine version, Info strings, exit code and duration of the validation period
- Tags: List of tags, with their comment, aggregation mode and a pointer to their Physical Units
- TagValues: Values of the tags for each run
- PhysUnit: Auxiliary table with the list of physical units
- ValiDatabaseVersion: version ID of the ValiDatabase

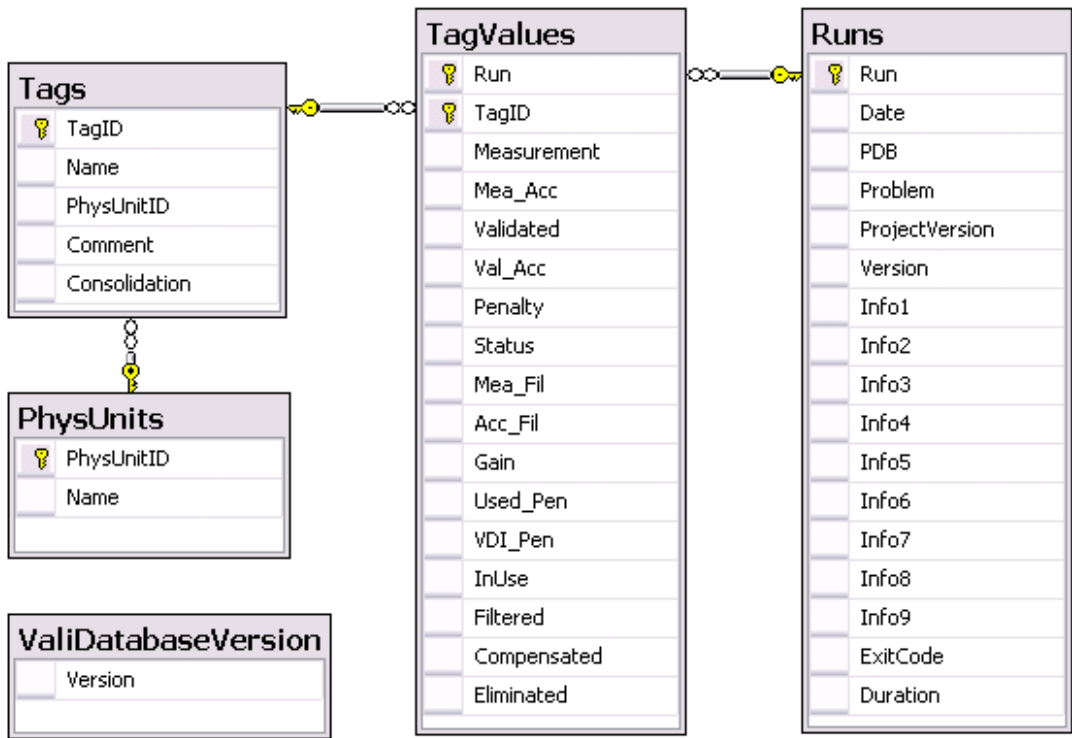


FIGURE 8 - VALI DATABASE STRUCTURE

SECTION 4: WORKING RUN BY RUN

If you have created flow-sheets with references to various Tag values, these values will always be updated with active Run (in the RUN work-sheet).

4.1 LOAD RUN DATA FILES INTO VALIREPORT

The RUN sheet will contain all the tags data related to a particular run or aggregated values related to runs between two selected dates. The data can be either be read from an archive file created by ValiEngine or from ValiDatabase (see section "Retrieve Run Stored in the Database").

Attention: Run data files are to be created using the ARCHIVE command in a measurement file, followed by the list of Tags to be included in the RUN or using "-ARC" Vali runtime option.

To read an archive file:

- select command Runs | Load Run or click  button on ValiReport toolbar
- select the arc file to be read

In addition to the tag values the RUN sheet also displays coloured bullets in front of the tag names. By default the bullet is green. It will be red if the corresponding stream or unit was OFF during the run. No bullet is displayed for pseudo tags like the number of equations or the name of the model file.

Attention:

1. The default directory where ValiReport will search for the run files can be modified in the Option dialogue (menu item Application | Options...)
2. The sheet "RUN" is protected, it's therefore not possible to make any change to the table.
3. Excel filters are activated on the top of each column, enabling to filter the data according to various conditions on the contents of the columns.
4. The bullet colouring is not displayed by default. To activate the option, select the command Application | Display | ON/OFF Status from the menu bar.

TAGNAME	UNITS	MEASURED	MEA.ACC	VALIDATED	VAL.ACC	PENALTY	STATUS	USED	USED.ACC	USED.PEN
%_ECONO	%	25.0612	-1E+20	24.6361	-0.841635	0	0	25.0612	-1E+20	0
%_REHEAT	%	12.4892	-1E+20	12.7791	-0.594429	0	0	12.4892	-1E+20	0
%_SUPHEAT	%	30.684	-1E+20	31.5875	-0.713041	0	0	30.684	-1E+20	0
%_VAPO	%	29.8244	-1E+20	29.0584	-0.95652	0	0	29.8244	-1E+20	0
1_P	bar	1.013	0	1.013	0	0	0	1.013	0	0
1_T	C	25	0	25	0	0	0	25	0	0
2_P	bar	1.013	0	1.013	0	0	0	1.013	0	0
2_T	C	25	0	25	0	0	0	25	0	0
AIR_0_HFLOWTOT	MW	-0.6722	-1E+20	-0.696933	33.8213	0	0	-0.6722	-1E+20	0
AIR_0_HUMIDITY	%	30.0014	-2	30.0073	-1.99952	9.72068E-05	0	30.0014	-2	9.72068E-05
AIR_0_MASSF	kg/s	306.08	-1E+20	305.202	-0.657625	0	0	306.08	-1E+20	0
AIR_0_MGO2	%	21	0	21	0	0	0	21	0	0
AIR_0_MOLF	Nm3/h	867783	-10	855209	-0.657867	0.0209954	0	867783	-10	0.0209954
AIR_0_P	bar	1.013	-2	1.0128	-2.00031	9.72717E-05	0	1.013	-2	9.72717E-05
AIR_0_T	C	15.1468	0.5	15.0867	0.493036	0.0144619	0	15.1468	0.5	0.0144619
AIR_1_T	C	70	0.5	70.0489	0.491224	0.00957464	0	70	0.5	0.00957464
AIR_2_P	bar	1.013	-1E+20	1.0128	-2.00031	0	0	1.013	-1E+20	0
AIR_2_T	C	339.642	2	338.321	1.79606	0.436515	0	339.642	2	0.436515
B-4_DP1	bar	1	0	1	0	0	0	1	0	0
BOILER_EFFICIENCY	%	95.6749	1E+20	95.6586	0.0748105	0	0	95.6749	1E+20	0
BURNER_DP	bar	0	0	0	0	0	0	0	0	0
CD.1_MASSF	kg/s	222.01	-1E+20	218.365	-0.476166	0	0	222.01	-1E+20	0
CD.2_HW	kj/kg	156.299	-1E+20	167.349	-0.593116	0	0	156.299	-1E+20	0
CD.2_MASSF	kg/s	0	-1E+20	218.365	-0.476166	0	0	0	-1E+20	0
CD.2_P	mbar	73	-2	73.6647	-1.3391	0.207285	0	73	-2	0.207285
CD.2_T	C	40.1	0.5	39.9543	0.237508	0.0848623	0	40.1	0.5	0.0848623
CD.3_P	bar	0.0635	-1E+20	0.0736647	-1.3391	0	0	0.0635	-1E+20	0
CD.3_T	C	37.19	1E+20	39.9543	0.237508	0	0	37.19	1E+20	0
CD.4_P	bar	2.594	-2	2.65377	-0.815299	1.32746	0	2.594	-2	1.32746

FIGURE 9 - THE SHEET RUN

For each tag the following data are available:

- TagName: name of the tag
- Units: physical units
- Measured: measured value
- Mea.Acc: accuracy of the measured value
- Validated: validated value
- Val.Acc: accuracy of the validated value
- Penalty: penalty calculated on basis of the measured value
Note: For a PFD, Penalty is the sum of Penalties of all the Tags defined in the PFD.
- Status: bit settings defining whether the tag is eliminated, flagged, filtered, etc.
 - Value 2 (2nd bit): eliminated or relaxed
 - Value 4 (3rd bit): filtered
 - Value 8 (4th bit): compensated
 - Value 16 (5th bit): BAD
 - Value 32 (6th bit): FLAT 1
 - Value 64 (7th bit): FLAT 2
 - Value 512: tag not used in this run
- Used: value used by the reconciliation engine, that if after filtering and compensation
- Used.Acc: accuracy used by the reconciliation engine
- Used.Pen: penalty calculated on basis of the used value
Note: For Tags, Used Penalty and Penalty are the same values. For a PFD, Used Penalty is the sum of Used Penalties of all the Tags defined in the PFD, except eliminated Tags.
- Used.Gain: gain of accuracy between the used and validated values
- VDI.Pen: penalty calculated using VDI formula on basis of the used value
- InUse: flag indicating whether the tag was actually used in the current run
- Filtered: flag indicating whether the tag was filtered
- Compensated: flag indicating whether the tag was compensated
- Eliminated: flag indicating whether the tag was eliminated by the gross error detection mechanism
- Aggregation: mode of aggregation (1=Average, 2=Begin date, 3=End Date, 4=Sum)
- Comment: tag comment

4.2 STORE RUN DATA INTO A DATABASE

To store a run into a ValiDatabase:

- select the command **Runs | Insert Run** or click 
- enter the date and time of the run. If a "RUNDATE" tag has been specified at the creation of the run file, it will be used by default. Otherwise, "DATE" and "TIME" tag will be used if present in the run file. If no such information tags are present in the run file, the default date and time will be the current one

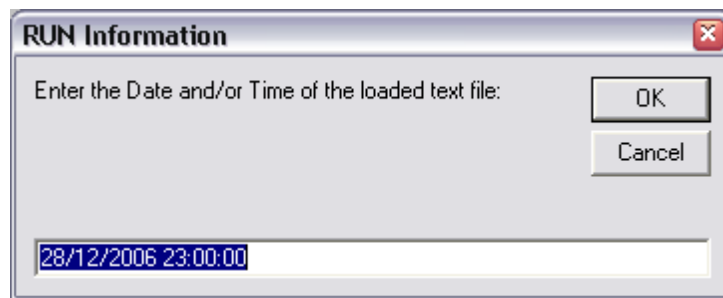


FIGURE 10 - GIVING A DATE AND TIME STAMP TO A RUN

4.3 AUTOMATICALLY STORE RUNS INTO A DATABASE

To automatically store runs in a ValiDatabase:

- select the command **Runs | Auto Insert Run....**

The auto insert Run stores ALL arc files from the default Input Files directory in the database without prompting the user. The date and time of the run must be specified in the arc file..

The way runs will be inserted in the database depends on the settings defined in the DB Settings page of the Application | Options notebook.

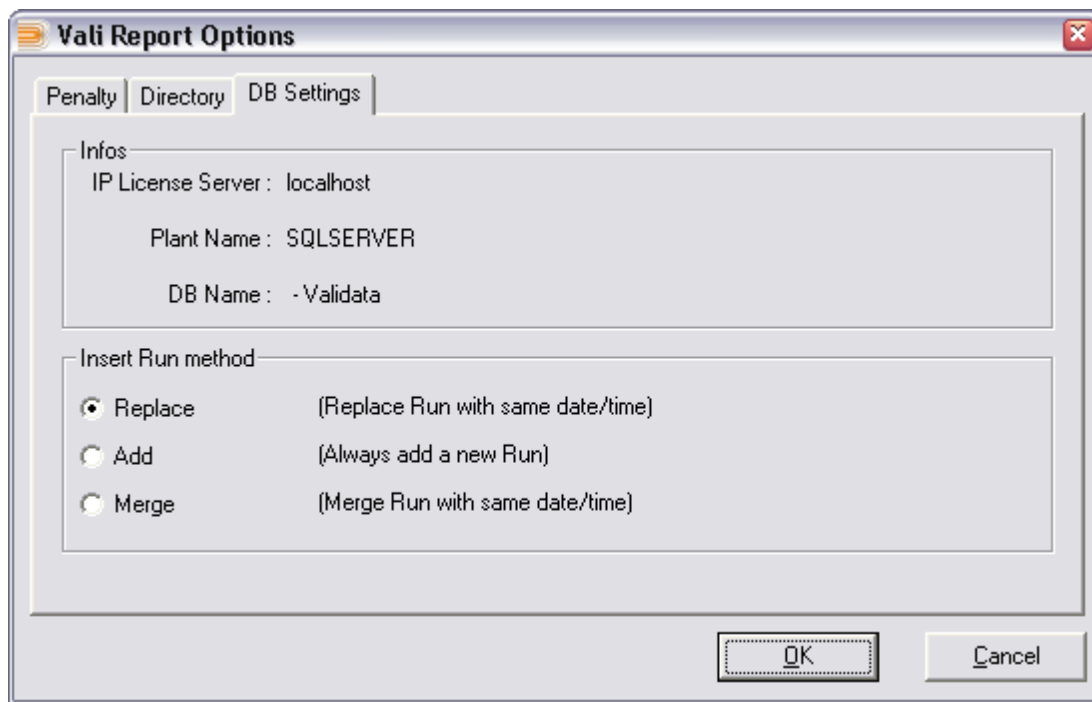


FIGURE 11 - DB SETTINGS

Replace:	any existing run with the same date and time will be removed and the new data will be put in place, reusing the highest ID run number.
Add	a new run is always created
Append	the data will be merged with the run with the same date and time and highest ID. If some tags are duplicated then their value will be updated with the new data.

4.4 RETRIEVE RUN STORED IN THE DATABASE

If you want to see the contents of a given run that is stored in the database:

- select the command **Runs | Retrieve Run...** or click  in the ValiReport toolbar
- select the desired run from the dialogue box, from the list of available runs for the active database and click **OK**

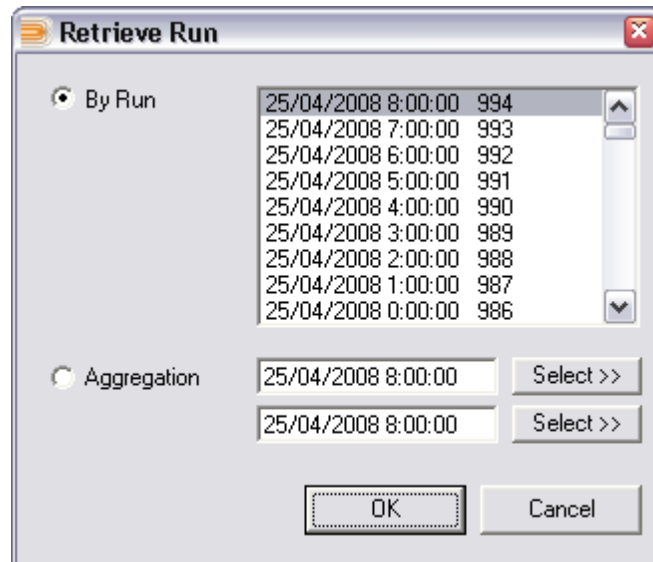


FIGURE 12 - RETRIEVING A RUN FROM THE DATABASE

Attention:

1. The runs are normally identified by a run number, which is unique in the database, and a date/ time that you have entered when storing it in the database.
2. The run continues to be stored in the database and all kind of changes do not affect the database contents. You can however choose the **Runs | Insert Run...** command from the menu bar to store it as a new run into the database.

You can also retrieve aggregated data from runs between two selected dates. The type of aggregation that will be made is specified in ValiModeller in the historian page of the tags.

When using the aggregation mode, the measured, used and validated values of the tags will be aggregated according to the user selected aggregation mode, the flags (filtered, compensated, ...) will be summed up (giving thus the number of runs in which the tag has been filtered, compensated, ...) and all other values (accuracies, penalties, gain) will be averaged.

4.5 REMOVING RUNS FROM THE DATABASE

To remove runs from the database:

- choose the **Runs | Remove Run...** command, which displays the dialogue box shown in figure below
- select the runs to be deleted and click **OK**

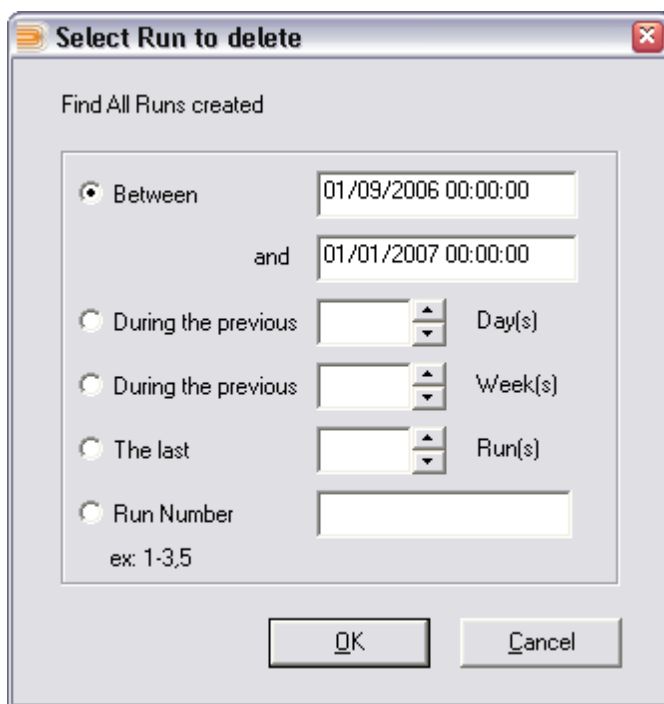


FIGURE 13 - DELETING RUNS FROM THE DATABASE

4.6 GROUPING TAGS BY UNITS

If you defined a structure file (see section Working Directory) an additional sheet called STRUCT will be created. It presents the same kind of information as the RUN sheet but the tags will be grouped by units according to the contents of the structure file.

Attention: The Struct sheet is not created by default. To activate the option, click on the ValiReport menubar: Application | Display | Struc Sheet. The Struct sheet will be created the next time you insert a run.

ValiReport - Coal.xls

File Edit View Insert Format Tools Data Window RCS Help Database Runs Application Vali Report Help

M11 2.74643

	A	B	C	D	E	F	G	H	I	J
1			RUN		12/06/2008 11:30:09					
2	TAGNAME	UNITS	MEASURED	MEA.ACC	VALIDATED	VAL.ACC	PENALTY	STATUS	USED	USED.ACC
3	DRUM									
4	DRUM_DP	bar	1	0	1	0	0	0	1	0
5	VAP.1_MASSF	kg/s	316.8	-1E+20	309.723	-1.07542	0	0	316.8	-1E+20
6	VAP.1_P	bar	177.1	-2	183.076	-0.595862	2.84687	0	177.1	-2
7	VAP.1_T	C	357	1	358.392	0.492969	1.9384	0	357	1
8	VAP.3_T	C	360	1E+20	358.392	0.492969	0	0	360	1E+20
9	VAP.3_P	bar	184	-1E+20	183.076	-0.595862	0	0	184	-1E+20
10	FW.10_MASSF	kg/s	270.9	-1E+20	271.291	-0.495529	0	0	270.9	-1E+20
11	FW.10_T	C	349.2	1	348.038	0.713214	1.34939	0	349.2	1
12	FW.10_P	bar	185.3	-2	184.076	-0.592625	0.109029	0	185.3	-2
13	BURNER									
14	BURNER_DP	bar	0	0	0	0	0	0	0	0
15	FUMES.1_T	C	2281	1E+20	2220.77	7.80517	0	0	2281	1E+20
16	FUMES.1_P	bar	1.013	-1E+20	1.013	0	0	0	1.013	-1E+20
17	FUMES.1_MFO2	%	2	-1E+20	2.03192	-4.52983	0	0	2	-1E+20
18	FUMES.1_MGO2	%	2.1	-5	2.14476	-4.50337	0.181696	0	2.1	-5
19	FUMES.1_MGSO2	ppm	640	-5	626.515	-3.5589	0.177575	0	640	-5
20	COAL_T	C	20	2	19.9921	1.99999	1.54141E-05	0	20	2
21	COAL_P	bar	1.013	0	1.013	0	0	0	1.013	0
22	COAL_WFC_COAL	%	92	-5	92.0093	-0.276609	4.11402E-06	0	92	-5
23	COAL_WFH_COAL	%	4.3	-5	4.27611	-4.84192	0.0123507	0	4.3	-5
24	COAL_WFS_COAL	%	0.89	-5	0.908318	-3.53782	0.169454	0	0.89	-5
25	COAL_WFH2O_COAL	%	2.8	-5	2.80625	-4.9724	0.00198997	0	2.8	-5
26	COAL_LHVWT	kJ/kg	35180	-2	35281.7	-0.527232	0.0208907	0	35180	-2
27	COAL_MASSF	t/h	74	-20	82.0271	-0.655768	0.294167	0	74	-20
28	COAL_HFLOWTOT	MW	0	-1E+20	826.638	-0.413086	0	0	0	-1E+20
29	AIR.2_T	C	339.642	2	338.321	1.79606	0.436515	0	339.642	2
30	AIR.2_P	bar	1.013	-1E+20	1.0128	-2.00031	0	0	1.013	-1E+20

Ready

FIGURE 14 - THE SHEET STRUCT

SECTION 5: TABLES AND CHARTS

With Tables and Charts you can display values of selected tags for a number of runs, either as Tables in worksheets or as Trend charts

5.1 CREATING A TABLE

To create a table, activate the menu command **Runs | Create Table and Chart** to open the Create Table and Chart dialog box. If a table already exists in the current sheet, you need to select the **New Sheet** option.

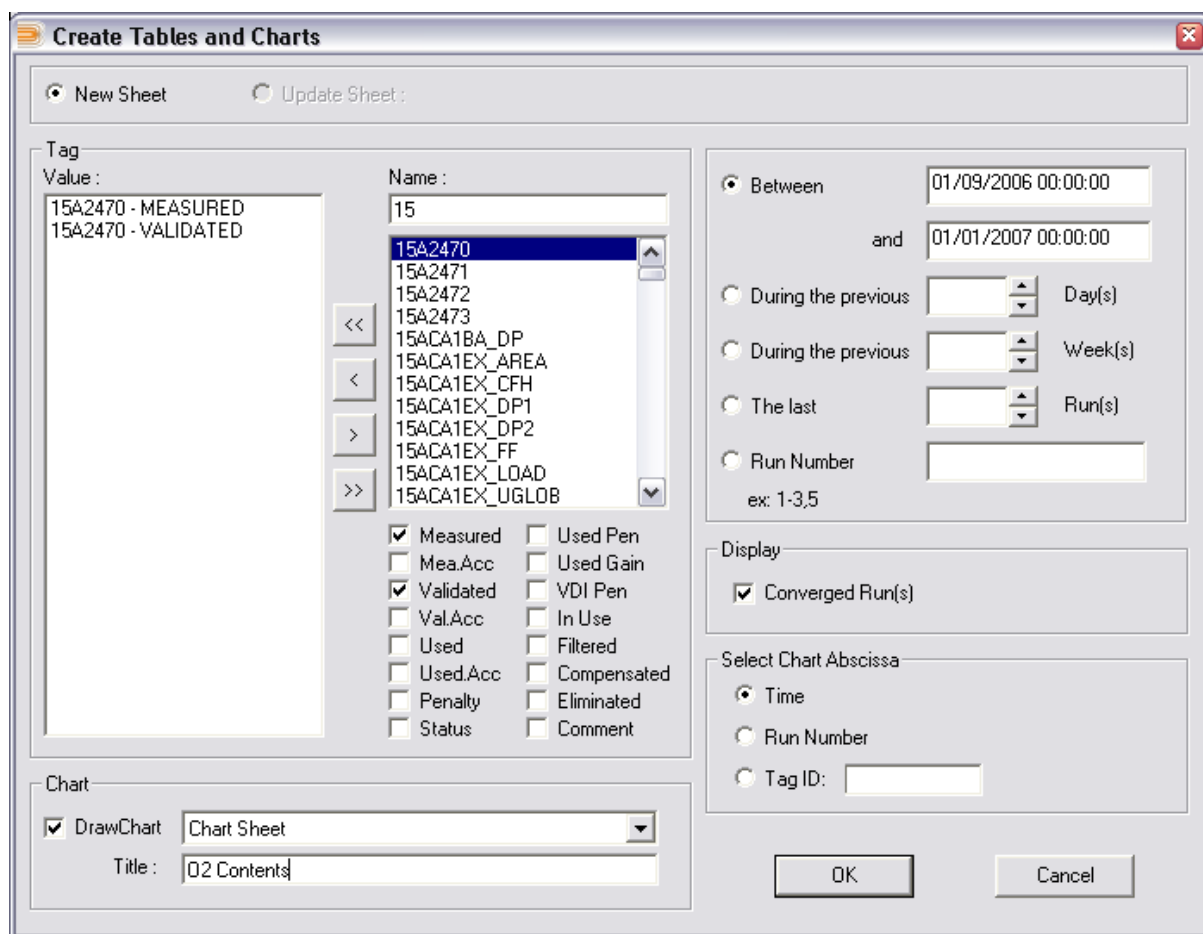


FIGURE 15 - TABLE AND CHART DIALOG BOX

In the left side of the dialog box you will select the tag values to be written in the table. You can use the mask field to limit the number of tags that are displayed in the selection box.

In the right side of the dialog box, you can select the runs in different ways. You can also request to select only converged runs avoiding polluting the data with inconsistent data

When you have selected the tag values and the runs click **OK** to generate the table.

5.2 CREATING A CHART

You create charts using the same dialog box as for creating Tables.

- select the **Drawchart** option at the bottom of the dialog box
- select whether you want your chart in a separate Chart sheet or embedded in the same sheet as the table
- define a **Title** for your graph

Attention: Once charts have been defined you can modify their appearance as you would do for any Excel chart.

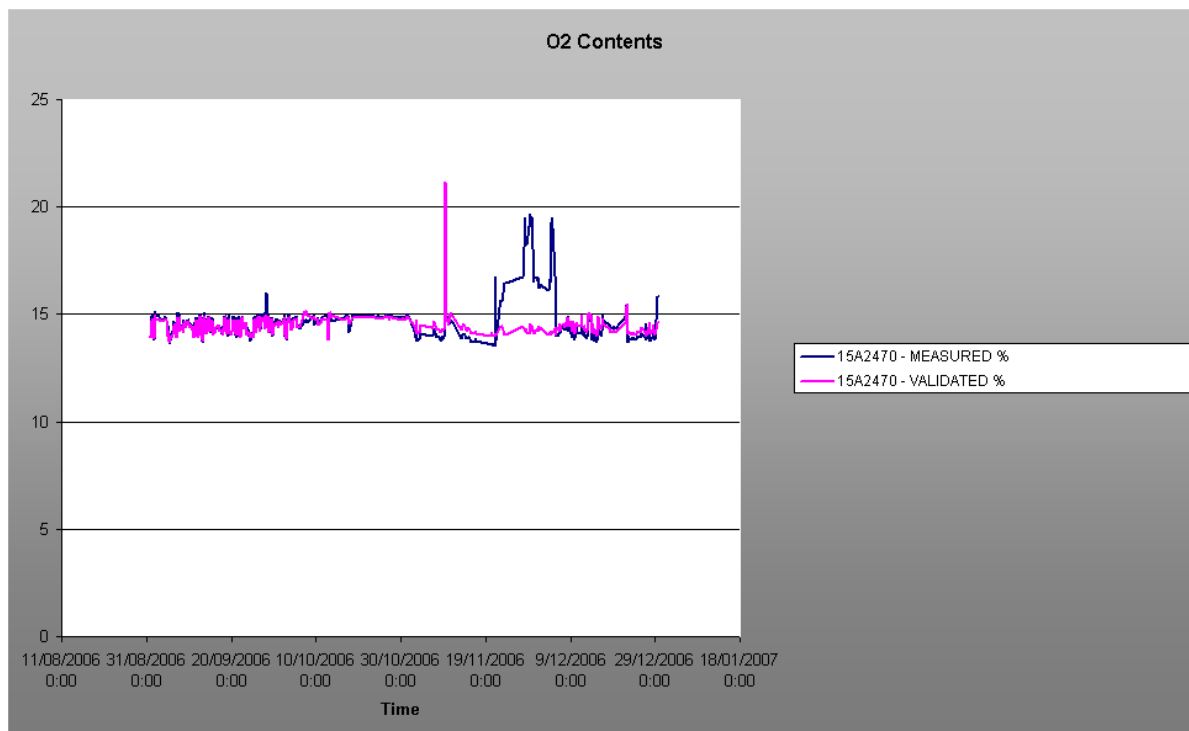


FIGURE 16 - A TYPICAL CHART

5.3 UPDATING TABLES AND CHARTS

To update a Table (and its associated charts), select the corresponding worksheet and click the **Runs | Create Table and Chart** menu command. The dialog box will open again with the option **Update Sheet** activated. You can modify the selection of runs and tags to be displayed in the Table.

SECTION 6: FLOW-SHEETS AND REPORTS

The application also enables you to design or import flow-sheets and customised reports.

Another feature is to change the colour of the cells containing suspect tags, i.e. with high PENALTY. If wanted, you can activate this feature only for some cells.

6.1 CREATING A NEW FLOW-SHEET

Add a new worksheet by right-clicking the mouse on the sheet tabs at the bottom of the sheet window. Select the **Insert...** item from the list and double-click on the worksheet's icon to create it. The worksheet is inserted at the left side of the active sheet.

Rename the worksheet, if you wish so, by double-clicking it on the sheet tabs at the bottom of the sheet window. You can name it for example as **Flowsheet1** to distinguish it from the worksheets that supports the charts.

6.2 DRAWING THE FLOW-SHEET

Use the drawing tools of Excel to draw the objects on your flow-sheet or import an existing drawing.

Display the toolbar by activating the check mark near **Drawing** in the dialogue box that appears when selecting the **View|Toolbars** command. You can then use the different design functions as you like.

If you already have examples of flow-sheets it is very easy to import them into ValiReport. However you must ensure that the flow-sheet has a *.wmf (Microsoft Windows MetaFile) extension. This file extension is, at the moment, the only way to add near your design objects tag references selected directly from the run in sheet RUN.

On the worksheet where you want to insert it select **Insert|Picture|From File...** and look, in the dialogue box, for the directory where it is stored.

Once the object is in the worksheet select **Format|AutoShape...**, choose the **Colors and Lines** pad and activate option button **No Fill** from **Fill** group.

Now you have, simultaneously, access to the flow-sheet and to the cells.

To have a better view of your flow-sheet it is possible to customise the columns size as well as the flow-sheet's zooming factor. Left-click and drag the right border on the columns header of the desired column to resize it. Change the zoom of the flow-sheet in the dialogue box that you activate by selecting **View|Zoom**.

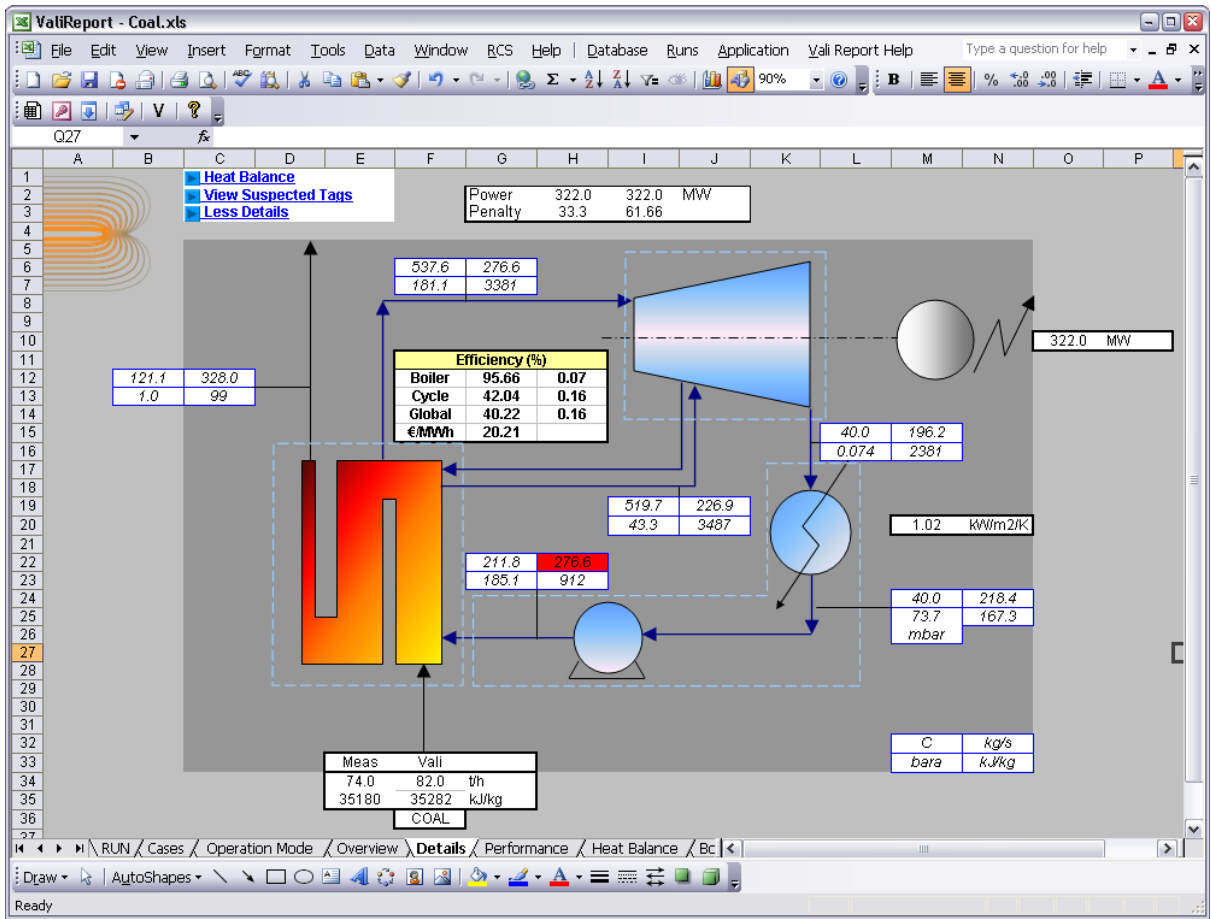


FIGURE 17 - A TYPICAL FLOW-SHEET

6.3 INSERTING TAG REFERENCES: FSTAG

The dialogue box is displayed by double-clicking a cell in the flow-sheet:

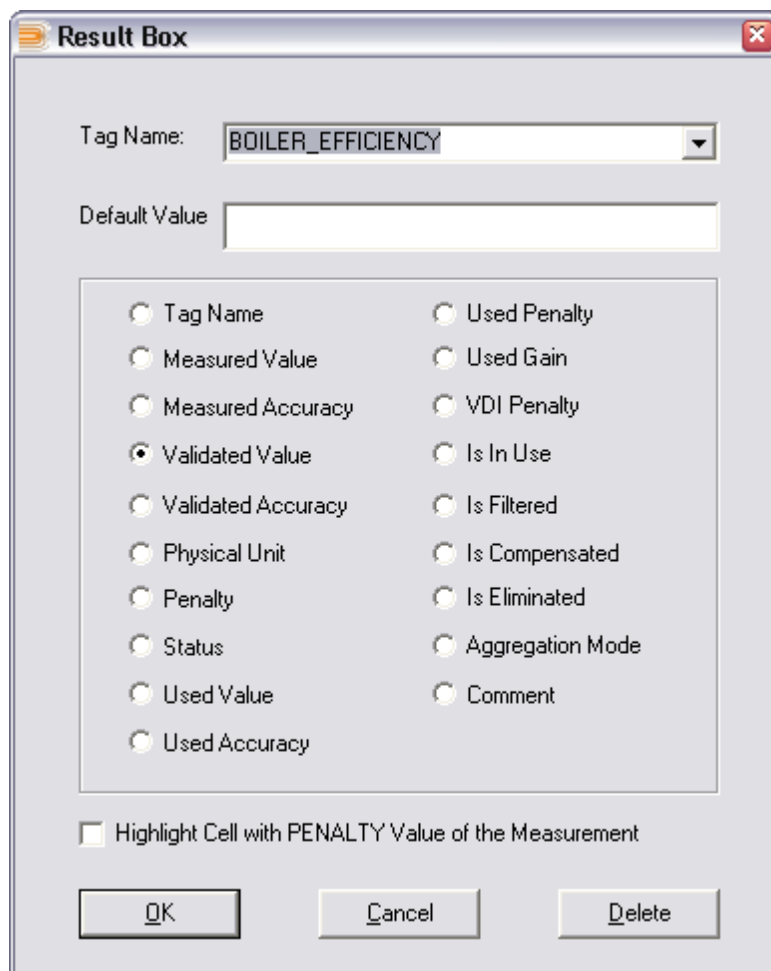


FIGURE 18 - WORK WITH A TAG REFERENCE ON A FLOW-SHEET

Select tag name from the drop-down list box and activate any of the types of values you want to select from the RUN sheet. Click **OK** and the value will be inserted in the flow-sheet's active cell.

If you also wish to mark the cells containing suspect tags, i.e. high **Penalty**, activate the check box near **Highlight Cell with PENALTY Value of the Measurement**. The cell's colour will change according to the selected colours and the threshold value in the Options Dialog from the item menu bar **Application|Options**.

The **default value** corresponds to the text displayed when ValiReport don't find a value for the corresponding tag.

The dialogue box will create a call to the function FSTAG in the corresponding cell:

FSTAG ("TAGNAME", **RUNS**, IDFIELD, IDCOLOUR, [DEFAULT VALUE])

where:

- "TAGNAME" : the name of the tag. Can refer an Excel cell (should not be enclosed in double quotes in this case)
- IDFIELD: field to display:
 - 1: tag name
 - 2: measured value
 - 3: measured accuracy
 - 4: validated value
 - 5: validated accuracy
 - 6: physical unit
 - 7: comment
 - 8: penalty
 - 9: status
 - 10: used value
 - 11: used accuracy
 - 12: used penalty
 - 13: used gain
 - 14: VDI penalty
 - 15: in use flag
 - 16: filtered flag
 - 17: compensated flag
 - 18: eliminated flag
 - 19: aggregation mode
- IDCLOUR: 0: no color highlight; 1: value with a high penalty will be highlighted
- DEFAULT VALUE: value displayed if no value is retrieved

6.4 INSERTING AGGREGATED TAG REFERENCES: BLSTAGC

An alternate function enables to display values aggregated over a number of runs:

BLSTAGC ("TAGNAME", "BEGINDATE", "ENDDATE", IDFIELD, "AGGREG", [CONVERSION, DEFAULTVALUE], where:

- "TAGNAME" : the name of the tag. Can refer an Excel cell (should not be enclosed in double quotes in this case)
- "BEGINDATE": begin date of the aggregation
- "ENDDATE": end date of the aggregation
- IDFIELD: field to display:
 - 1: tag name
 - 2: measured value
 - 3: measured accuracy
 - 4: validated value
 - 5: validated accuracy
 - 6: physical unit
 - 7: comment
 - 8: penalty
 - 9: status
 - 10: used value
 - 11: used accuracy
 - 12: used penalty
 - 13: used gain
 - 14: VDI penalty
 - 15: in use flag
 - 16: filtered flag
 - 17: compensated flag
 - 18: eliminated flag
 - 19: aggregation mode
- "AGGREG": aggregation procedure:
 - **BEGIN**: value at begindate
 - **END**: value at enddate
 - **SUM**: sum of all values between begindate and enddate
 - **AVG**: average of all values between begindate and enddate

- **COUNT:** count of number of values between begindate and enddate
- **CONVERSION:** an optional multiplying factor
- **DEFAULTVALUE:** an optional default value
- **CRITERIA:** an optional criteria on the selected field (Example: "> 4" will select only values above 4)

Example:

```
=blstagc("14A2470";"1/9/2006";"1/10/2006";2;"AVG")
```

will calculate the average of tag 14A2470 between the 1st September and the 1st October 2006.

6.5 MODIFYING TAG REFERENCES

If you want to change a FSTAG tag reference, double-click on it and make the desired changes on the displayed dialogue box.

Alternatively you can edit the contents of the cell directly. This mode also applies for BLSTAGC function.

6.6 DELETING TAG REFERENCES

To delete a tag reference, click simply the **Delete** button on the dialogue box.

6.7 CHANGING COLOURS AND THRESHOLD VALUE

By default, the application assigns no colour to the cell if its **PENALTY** is less than 4, and red if it is bigger than or equal to 4. The user can modify these default colours and the threshold value in the Option dialog from the item menu bar **Application | Options**.

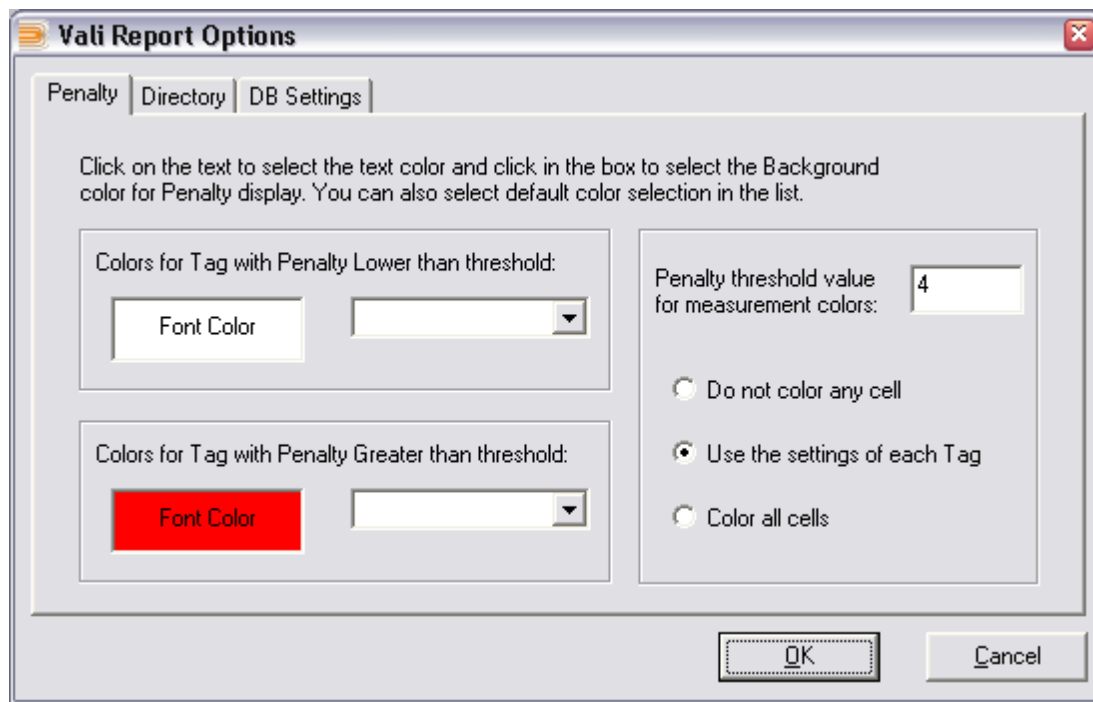


FIGURE 19 - PENALTY COLOURS OPTIONS

Each colour can be manually defined by clicking on the background and/or on the text of the boxes. A set of predefined couple of colours can be selected in the drop down boxes.

The threshold value defined the colours of an ValiReport cell by comparing the penalty of the measurement with that value. You can also toggle OFF the colouring of all cells or colour every cells with the penalty value whatever you selected for each measurement display.

SECTION 7: WORKING DIRECTORY

ValiReport application use default work directories that can be customised.

ValiReport application uses three working directories:

- DataBase: folder containing the Access databases
- Archive: unused at this time
- InputFiles: folder containing the run data files (arc files)

To change the working directories, use the menu bar item **Application | Options**.

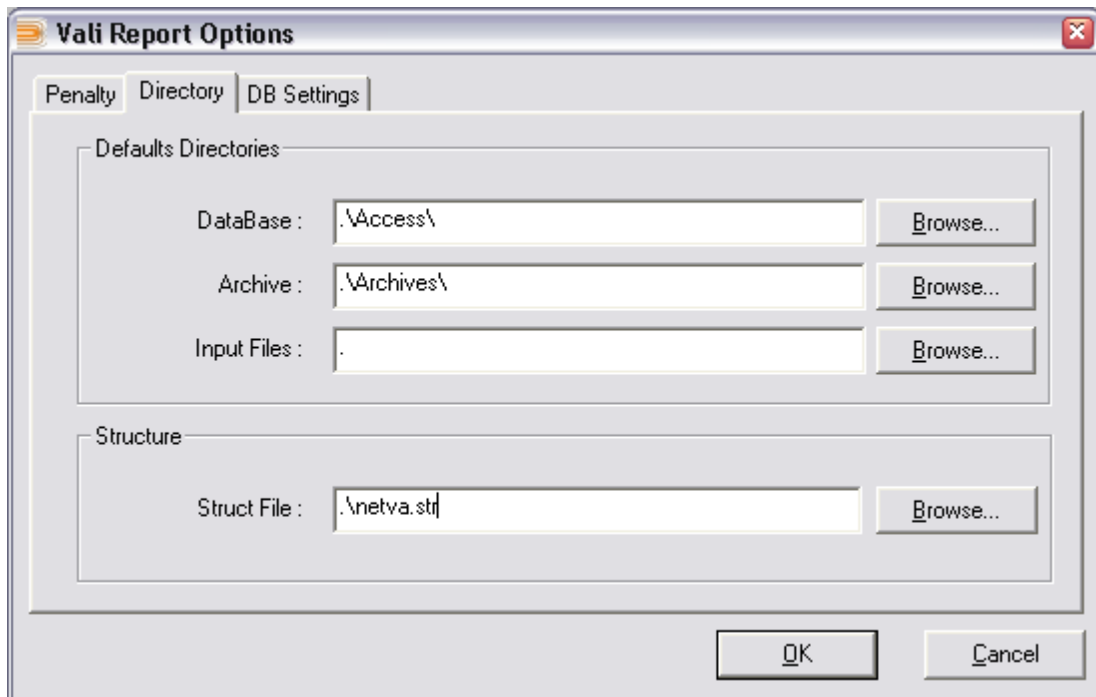


FIGURE 20 - DIRECTORY OPTIONS

All databases must be in the respective working directory. If you change the working directory, the databases of the former directory will not be accessible from ValiReport interface.

The input files (with the termination "-arc") should be placed in the corresponding directory. When the user load an "arc" file from the interface, he can however browse for another directory. On the other hand, the ValiDBLoader application doesn't prompt any question to the user and only look for available information in that directory.

The **structure file** is used to group the tags in the **Struct** page (see sub-section Groupings Tags by Units).

SECTION 8: VALIDBLOADER APPLICATION

ValidBLoader application is located in the install directory of ValiReport. It performs the same tasks as the Auto Insert feature (see section 3, menu item **Runs | Auto Insert Run**). ValidBLoader has been designed for online operation and can easily be automated via ValiScheduler. It receives as parameter the name of an ini file (e.g. ValidBLoader.ini) describing where the run data files are located, the name of the database, etc.

Attention: You can use the ini file associated to your ValiReport as ini file for ValidBLoader

ValidBLoader reads all run data files (*.arc) and stores the data into ValiDatabase to be exploited through ValiReport. The file ValidBLoader.log contains the log of the operations

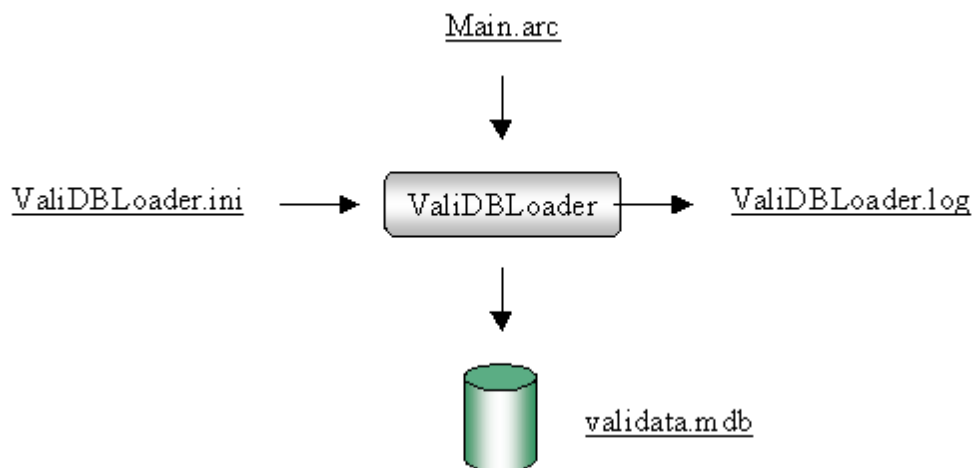


FIGURE 21 - VALIDBLOADER

Alternatively you give as second parameter the name of an arc (or mea) file to ValidBLoader, in which case only the designated file will be loaded into the ValiDatabase.

8.1 INI FILE FOR AN ACCESS DATABASE

```
[LicenseServer]
Url=localhost
```

URL of the Belsim LicenseServer

```
[Parameters]
TypeDataBase=ACCESS
Replace=DBReplace
```

Type of DataBase
DBReplace, DBMerge or DBAdd

```
[Directory]
Input File=D:\Projects\ValiReport_MR\InputFiles\
```

Input Files folder

```
[ACCESS]
Directory=D:\Projects\ValiReport_MR\Access\
DataBase=test.mdb
```

Database Folder
Database Name

8.2 INI FILE FOR A SQL SERVER DATABASE

```
[LicenseServer]
Url=belsim08
```

URL of the Belsim LicenseServer

```
[Parameters]
TypeDataBase=SQLSERVER
Replace=DBReplace
```

Type of DataBase
DBReplace, DBMerge or DBAdd

```
[Directory]
Input File=D:\Projects\ValiReport_MR\InputFiles\
```

Input Files folder

```
[SQLSERVER]
PlantName=VALIREPORT_OL
DataBase=VALIREPORT_OL
```

Plant Name in LicenseServer
Database name

8.3 ADDITIONAL SECTIONS IN THE INI FILE

You can now change the behavior of the program regarding the optional second argument by adding the following section in the INI file:

[SECOND ARGUMENT]

StoreArcFile=0	Arc File is renamed (1) or not (0)
StoreMeaFile=0	Mea File is renamed (1) or not (0)

If there is no second argument:

[MULTIPLE LOAD]

LoadArcFiles=1	All Arc Files will be loaded (1) or not (0)
LoadMeaFiles=0	All Mea Files will be loaded (1) or not (0)
StoreArcFiles=1	Arc Files are renamed (1) or not (0)
StoreMeaFiles=0	Mea Files are renamed (1) or not (0)

Log system:

[LOGFILE OPTION]

Path=	Path of the log file
File=	Name of the log file
LogLevel=INF	Error Messages Level
Backup Option=BACKUP	
MAXSIZE=10	Maximum size of the log file (in MB)

Dictionary MEA file:

Some values that must be interpreted by the program are not constant (in a MEA file). So, a dictionary file can be specified to interpret these values.

The path can be absolute or relative to the ValiDBLoader INI file:

[DICTIONARY]

Path=Dictionary\Dictionary.mea	Path of the dictionary file

Production Accounting:

For Production Accounting applications, the product movements data can be stored in dedicated tables.

[PRODUCT]	
LoadProduct=	Product movements data will be loaded (1) or not (0)
LogDirectory=	Path of the log file specific to Production Accounting
LogFile=	Name of the log file
BlsFile=	Name of the VALI4 model file with its path . ValiDBLoader will retrieve the Areas, Movements and Sub-Movements from the model file.
ProductDictionary=	The Products.dic file with its path . ValiDBLoader will retrieve the list of products from this dictionary file.

8.4 VALIDBLOADER STATUS

At the end of its execution, ValiDBLoader produces a file named **ValiDbLoader_log.ini** containing the keyword EXITCODE followed by a numeric code:

- -1: a fatal error did occur
- 0: ValiDBLoader was successful

Example:

```
[EXITCODE]
ExitCode=0
```

SECTION 9: RUNNING VALI

From ValiReport you can also run ValiLink and ValiEngine

To activate this option use either the menu command Application|Run Vali or press the  button from the toolbar. This opens a new Notebook where you can specify the Vali runtime options.

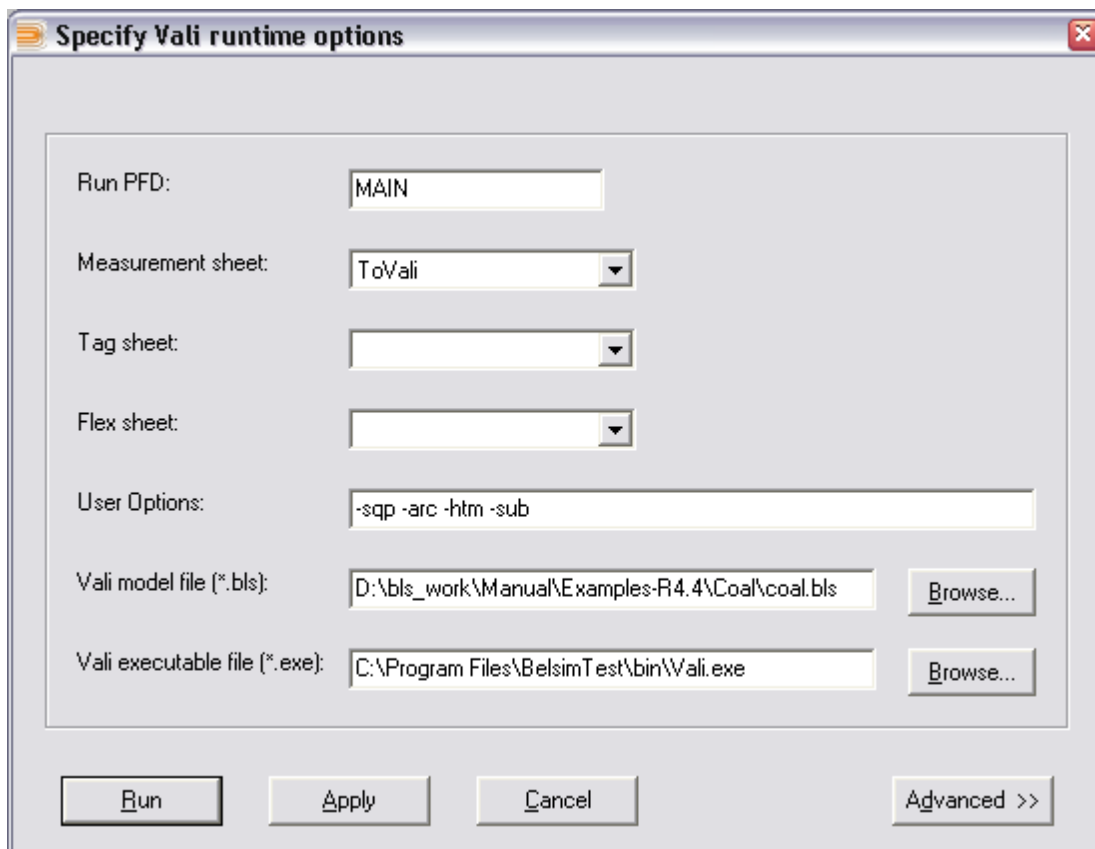


FIGURE 22 - VALI DIALOG BOX

In this page you specify the settings of VALI (written in the pc.vif file):

- which PFD of your model must be run,
- the names of the Measurement, Tag and Flex sheets: each of these sheets will be written under the name <sheet1>.mea, <sheet2>.tag and <sheet3>.rea and will be passed over to VALI,
- the runtime options (see [Running the model](#)),
- the name and location of the VALI model file,
- the name and location of the VALI executable.

Clicking on "Run" button will execute VALI once and will use a ValiOffline license.

To access the advanced feature click on the button "**Advanced >>**"

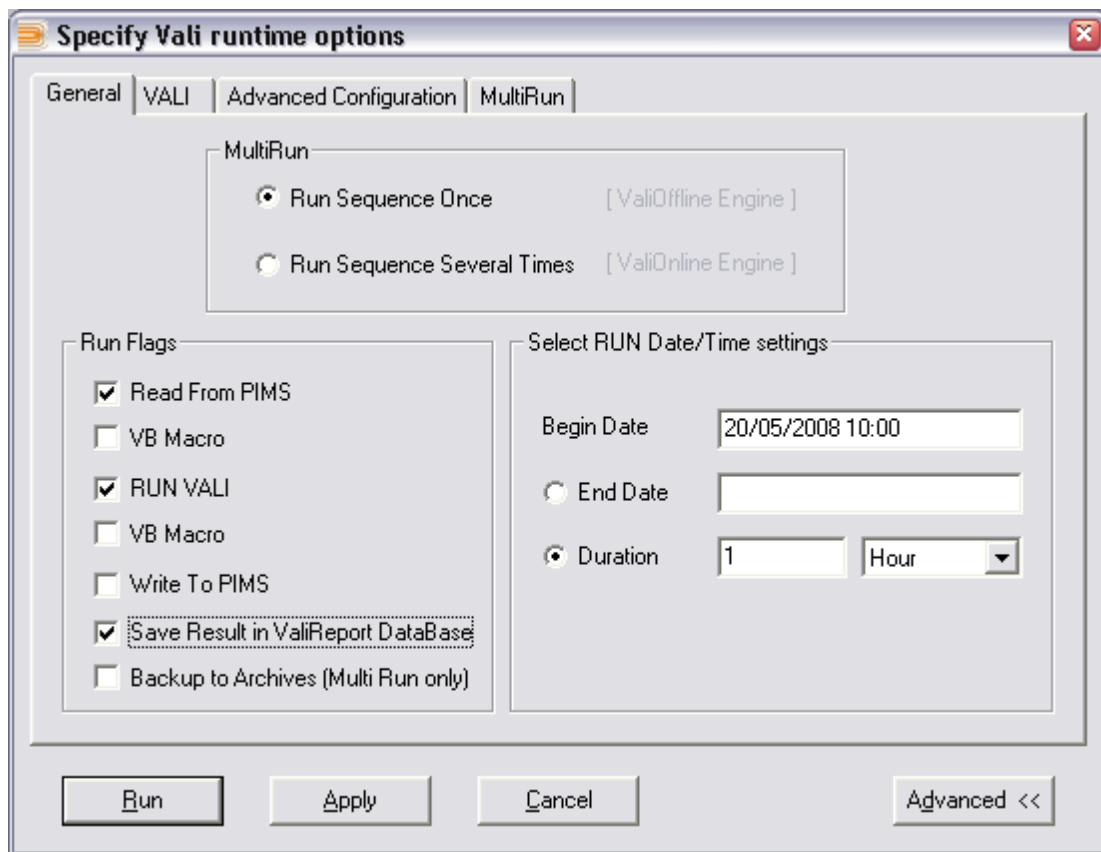


FIGURE 23 - ADVANCED VALI DIALOG BOX

The **Advanced** page is used to specify:

- whether to run VALI once (default option) or let Excel execute VALI several time in a row
- which steps are to be performed:
 - reading from a PIMS (using an external tool like ValiLink),
 - executing a Visual Basic Macro before running VALI
 - running VALI,
 - executing a Visual Basic Macro after running VALI
 - writing back result into a PIMS (using an external tool like ValiLink),
 - importing and saving the results in ValiReport database (ValiDatabase).
 - Making a back-up of the working directory
- the Begin Date and either End Date or Duration period to be considered for the Validation. These settings are used by ValiLink application,

Executing VALI once uses a ValiOffline Engine license.

Executing VALI several times uses a ValiOnline Engine license. See the **MultiRun** tab for detailed

configuration explanation.

Clicking on the button "**Advanced <<**" will send you back to the non advanced settings.

In the page **VALI** you specify the settings of VALI (written in the pc.vif file):

- which PFD of your model must be run,
- the names of the Measurement, Tag and Flex sheets: each of these sheets will be written under the name <sheet1>.mea, <sheet2>.tag and <sheet3>.rea and will be passed over to VALI,
- the runtime options (see [Running the model](#)),
- the name and location of the VALI model file,
- the name and location of the VALI executable.

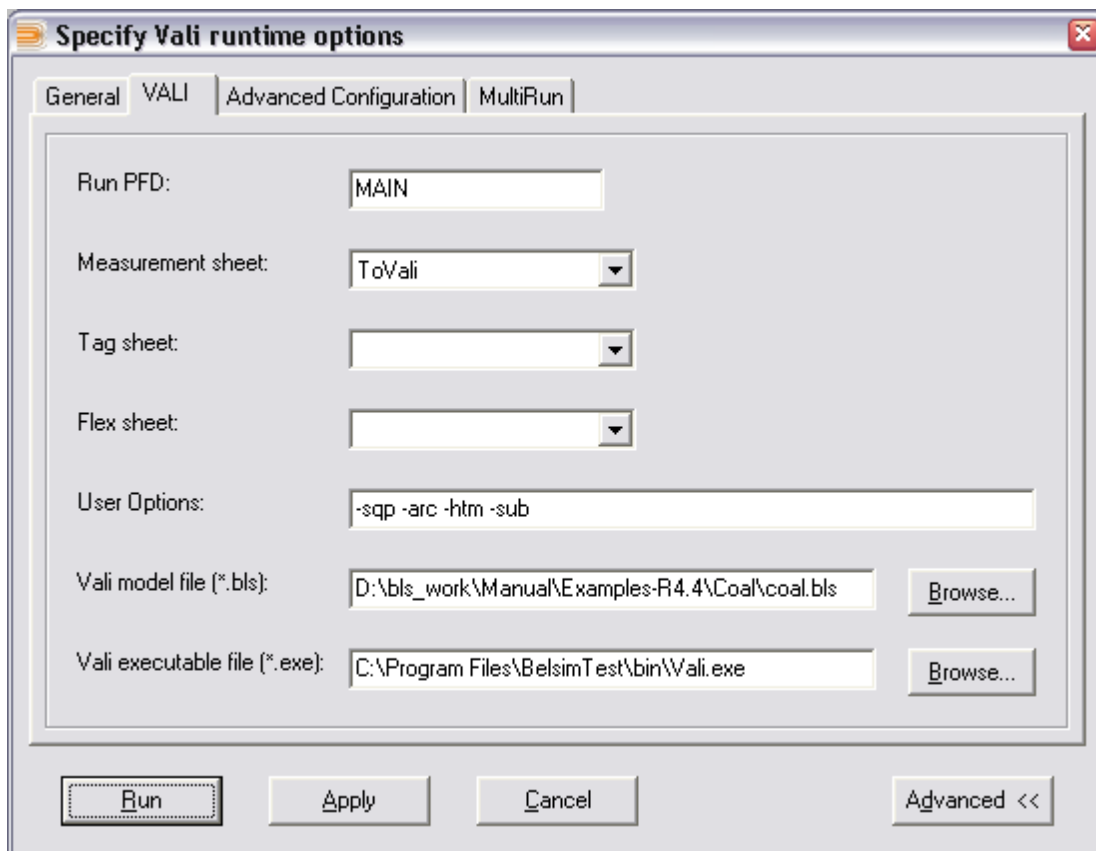


FIGURE 24 - VALI PAGE OF THE VALI DIALOG BOX

In the **Advanced Configuration** page you specify the settings for ValiLink :

- the name of the INI file for reading from the PIMS,
- the name of the INI file for writing to the PIMS,
- the name and location of ValiLink executable.
- The Visual Basic macro to execute before VALI
- The Visual Basic macro to execute after VALI

The syntax to specify a macro is:

- **MyVBMacro** to execute a macro defined in the ValiReport Workbook
- **MyWorkbook!MyVBMacro** to execute a macro defined in another Excel Workbook named "MyWorkbook":

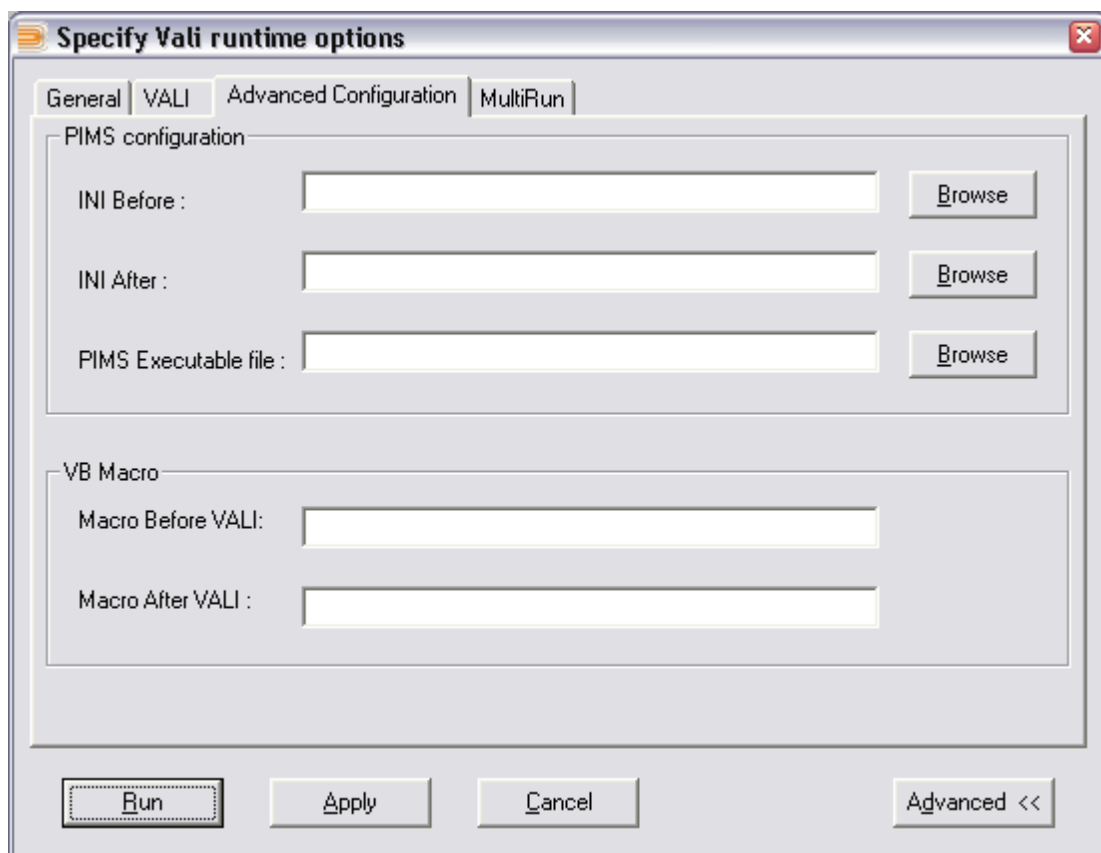


FIGURE 25 - ADVANCED CONFIGURATION OF THE VALI DIALOG BOX

In the **MultiRun** page you specify the settings to execute VALI several times in a row:

- First Run [input]: the number ID of the first run, can be linked to an excel cell
- Last Run [input]: the number ID of the last run, can be linked to an excel cell
- Current Run [output]: the number ID of the current run, must be linked to an excel cell

VALI will be processed several times starting from the First Run ID to the Last Run ID by increment

of 1. The First Run and Last Run can be specified directly in the text box or linked to an Excel cell. The current run ID will be printed in the linked cell to enable dynamic behaviour of the measurement sheet.

Example:

- First Run = 5
- Last Run = 11
- => Current Run will have the values 5, 6, 7, 8, 9, 10 and 11

The syntax to link a text box to a cell is:

- **C2** for the cell located in the second row and third column if you link the text box to a cell of the Excel sheet used as measurement file
- **Sheet1!C2** for the cell located in the second row and third column if you link the text box to a cell of the Excel sheet "Sheet1":

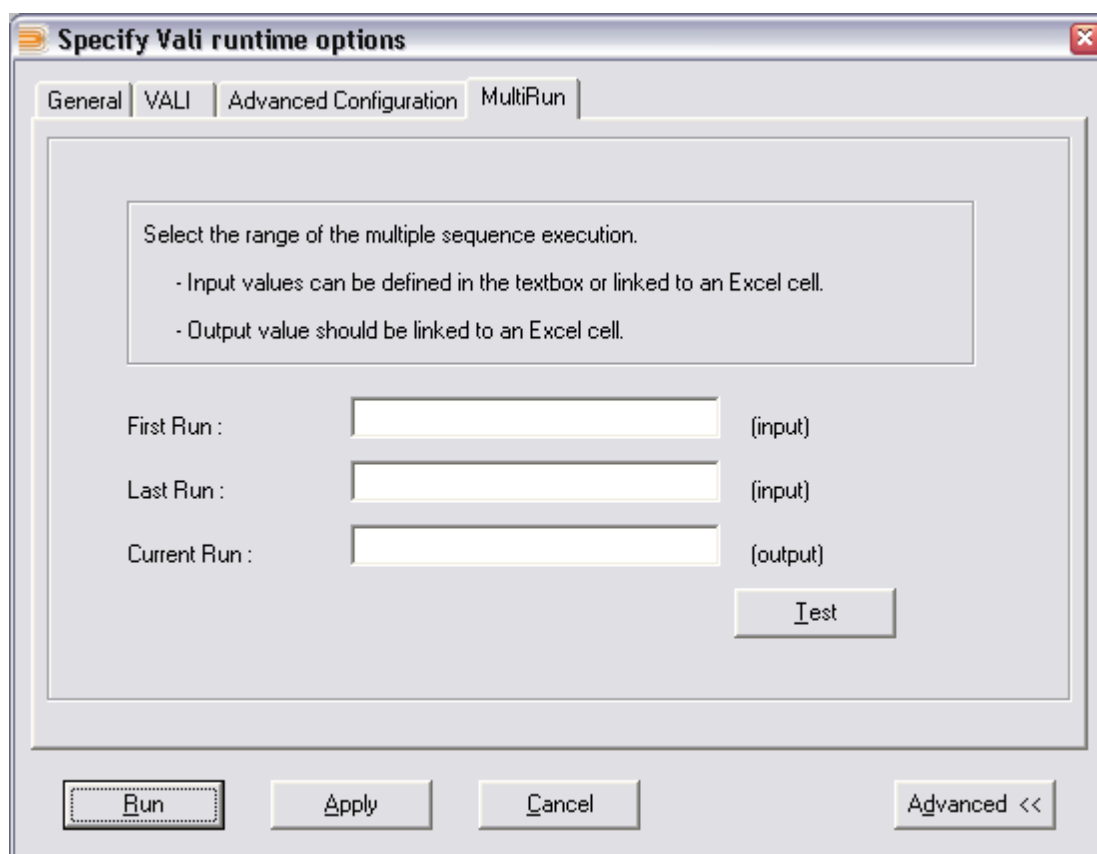


FIGURE 26 - MULTIRUN

Attention: The Multi Run feature uses a ValiOnline Engine.



belsim

CHAPTER 10

VALIANALYTICS

SECTION 1: INTRODUCTION

ValiAnalytics is an easy to use graphical user interface for sensitivity analysis of your results and your model. All results for the sensitivity analysis can be accessed through this interface, there is a graphical view of the results. Navigation between related tags is also easy in this interface. In addition, general information about the model can be found in this interface.

ValiAnalytics provides important information like:

- The error detectability of measurements, which is the maximum error that can occur on a measured value before the gross error mechanism can detect and remove the error
- The maximum bias of measurements, which is the maximum error that will occur on the reconciled value of the variable just before the error detectability limit for the corresponding measurement is reached
- The robustness of KPIs, which is an indication whether errors on measured values will or not be detected before some given KPI is biased by an amplitude exceeding its uncertainty

ValiAnalytics can be found in the bin sub-directory of Belsim installation directory. It can also be opened directly from ValiModeller, using the ValiAnalytics button in the toolbar.

ValiAnalytics offers two different views to display your results:

1. A global view, displaying general results about your model, such as number of redundancies, value of chi-square test, a distribution of gain on tags in your model, average gain on each measurement type, etc.

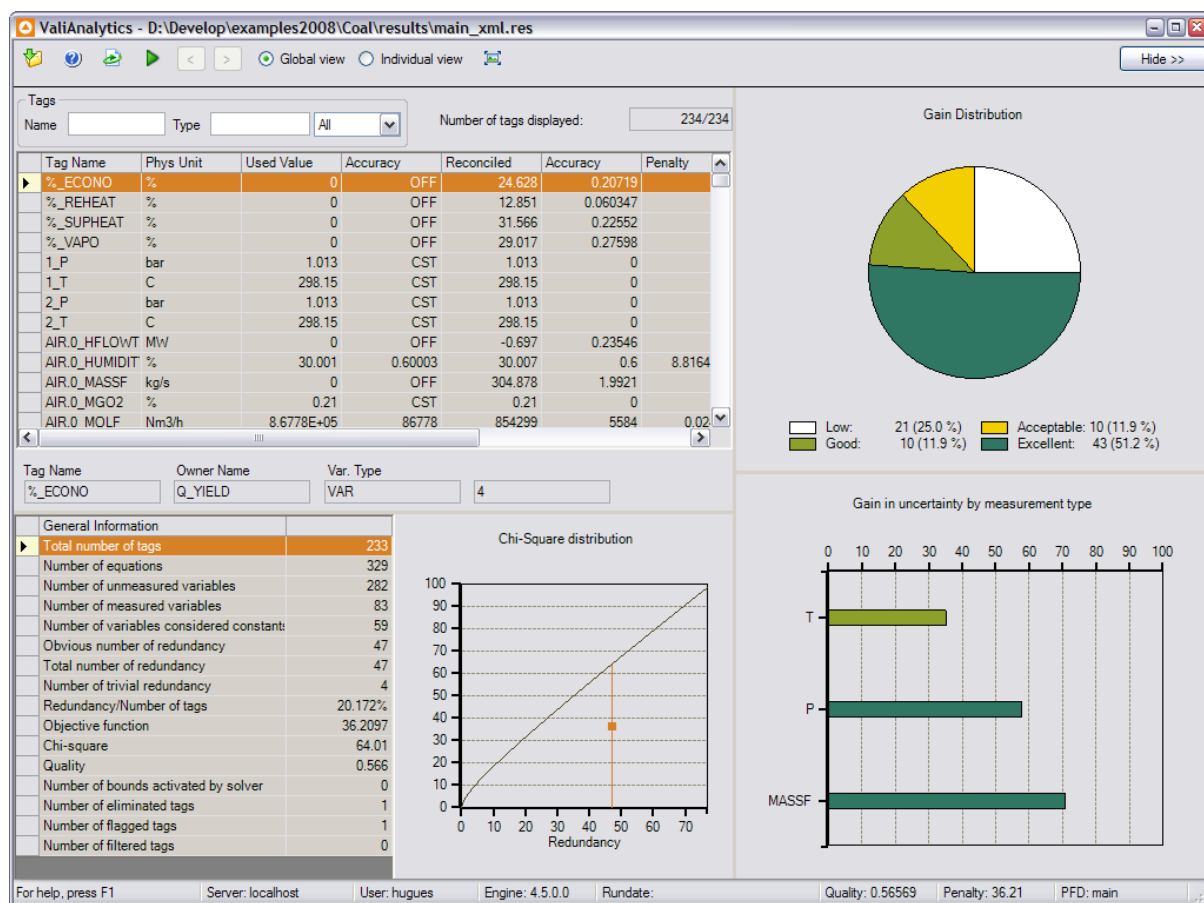


FIGURE 1 - GLOBAL VIEW

- An individual view, showing information about a selected tag, such as Gaussian distribution of uncertainty, list of contributors to uncertainty, list of tags dependent on selected tags, etc.

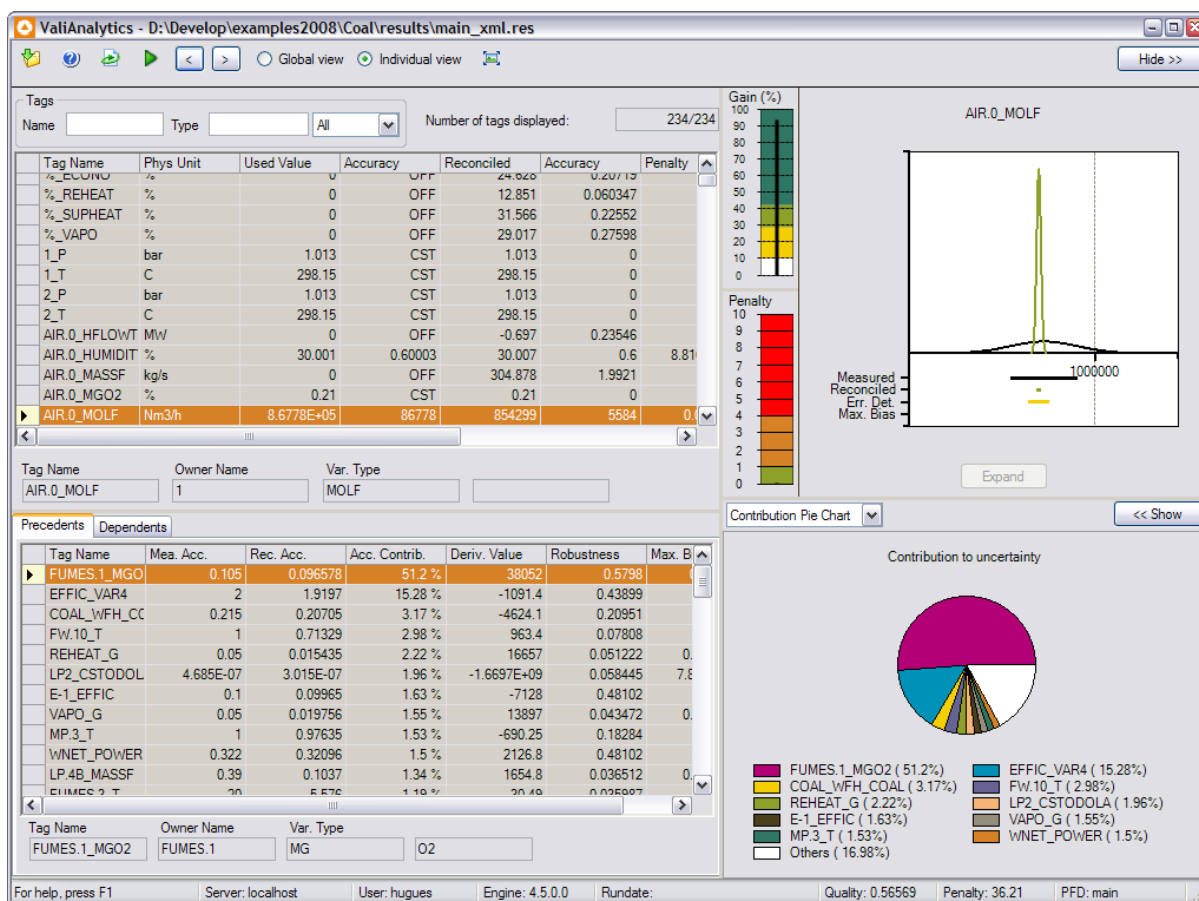


FIGURE 2 - INDIVIDUAL VIEW

SECTION 2: INTEGRATION IN VALISUITE

ValiAnalytics displays sensitivity analysis results from the sensitivity analysis program, that can be found in bin directory of VALI installation. Details on how to use this program can be found in [Sensitivity analysis help](#).

When processing runs, ValiEngine produces a file called <main>.fmt where <main> is the name of the pfd or unit which was last run, and containing various information about current run.

If Vali is run with option -SNS (by checking "Sensitivity report"), this file will be processed by the sensitivity analysis program, which will calculate sensitivity results for each measured value. Variables set as constants will be shown, but influence of these variables is not shown (impact of constants can be obtained by setting parameter "Constants" to "are set as measurements" in unit OPTVAL). The sensitivity analysis program can also be used as a stand-alone program. For more information, please refer to [Sensitivity analysis help](#).

The sensitivity analysis program produces a file called <main>_xml.res containing all sensitivity analysis results, as well as some general information about current run (number of measurements, redundancies, value of chi-square test...). Definition of the different information produced for each tag can be found in the [Tags table description](#). Definition of the relation between related tags can be found in the [Precedents/dependents tables description](#). Be aware that this file can be quite big.

Path between the different applications is shown below:

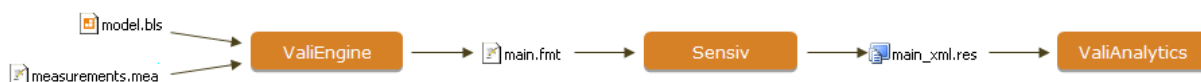


FIGURE 3 - FROM VALIENGINE TO VALIANALYTICS

SECTION 3: GENERALITIES

This section gives general information about ValiAnalytics, and serves as a getting started guide.

Covered topics are:

- [toolbar](#)
- [status bar](#)
- [general functionalities](#)

3.1 TOOLBAR

The toolbar is located at the top of the window and can be used to access ValiAnalytics main functionalities.

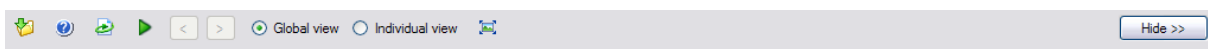


FIGURE 4 - TOOLBAR

Hovering the mouse on any of the button will display a tooltip describing the function of this button.



OPEN FILE

To open a file, simply click on the open file button. Clicking this button will open a file browser window, allowing you to open sensitivity analysis results file (<main>_xml.res). In the case where ValiEngine was last executed without sensitivity analysis selected, a prompt window will ask you if you wish to execute a sensitivity analysis for this later run.

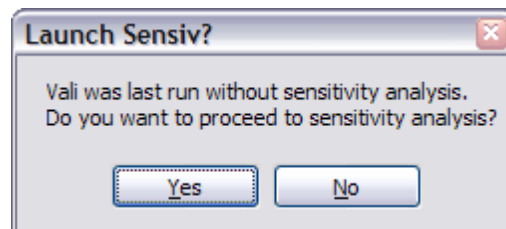


FIGURE 5 - VALI RESULTS MORE RECENT THAN SENSITIVITY RESULTS

You can open a file by entering a command-line argument to ValiAnalytics.



OPEN HELP

The open help button will open this help. You can also press the F1 key to display the help at any time.



RELOAD FILE

The reload file button allows you to reload the current file, i.e. to read the last version of the file, for example in the case where you would execute a sensitivity analysis outside of ValiAnalytics. In that case there is no need to browse to find the file again. An error message will be displayed if no file has been opened yet.



PROCEED TO SENSITIVITY ANALYSIS

This button launches the sensitivity analysis program. If no file is opened in ValiAnalytics, you will be requested to browse to the directory where the model file is located. If a file is opened, sensitivity analysis results will be recalculated for the current file.



PREVIOUS/NEXT TAG

These buttons enable navigation between previously selected tags. These buttons are disabled when first starting the application, as no tag has been previously selected. Button previous will be enabled once a second tag is selected; button next will be enabled after button previous has been clicked.



Global view



Individual view

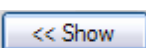
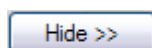
SWITCH VIEW

ValiAnalytics offers two views. Views can be switched by selecting desired view in the toolbar.



FIT CONTENT

The different parts of the ValiAnalytics window can be resized independently, allowing you to focus on the zone you're interested in. By pushing the fit content button, the different parts of the window will be automatically resized to divide the screen in four equal parts.



HIDE/SHOW BUTTON

This button, located at the extreme right of the window, allows to show or hide the right hand part of the window, offering the possibility to have a better overview of tags table. This button's text changes according to the action performed when clicked.

3.2 STATUS BAR

The status bar displays general information about the current run, as well as the name of the current user and the name of the License Server to which you are connected.

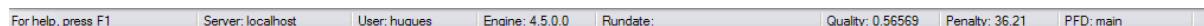


FIGURE 6 - STATUS BAR

The different elements that are displayed in this status bar are (in order, from left to right):

- Current status of application, i.e. if application is loading file ("Loading...") or idle ("For help, press F1")
- IP address of the License Server to which application is connected
- Name of current user
- Version of sensitivity analysis program (Sensiv.exe) used
- Date and time of run
- Quality of results (value of penalty divided by value of chi-square)
- Penalty of run (objective function)
- Name of unit or pfd that was processed (<main>)

3.3 GENERAL FUNCTIONALITIES

ABOUT TABLES

Tables can be customized in ValiAnalytics. By right-clicking on a table, you will get access to a list of optional columns that can be displayed. The options you choose are saved in <user>.ini file, and are reused each time you open ValiAnalytics.

By right-clicking on tables, it is also possible to export displayed results to Excel, and to print current table.

ABOUT CHARTS

All presented charts can be viewed in an external window. To open this window, simply double click on a chart. This operation opens a second window, presenting the chart in bigger size, and enabling to save this chart in selected format. Saved file will be located next to sensitivity analysis results (same folder as <main>_xml.res file).

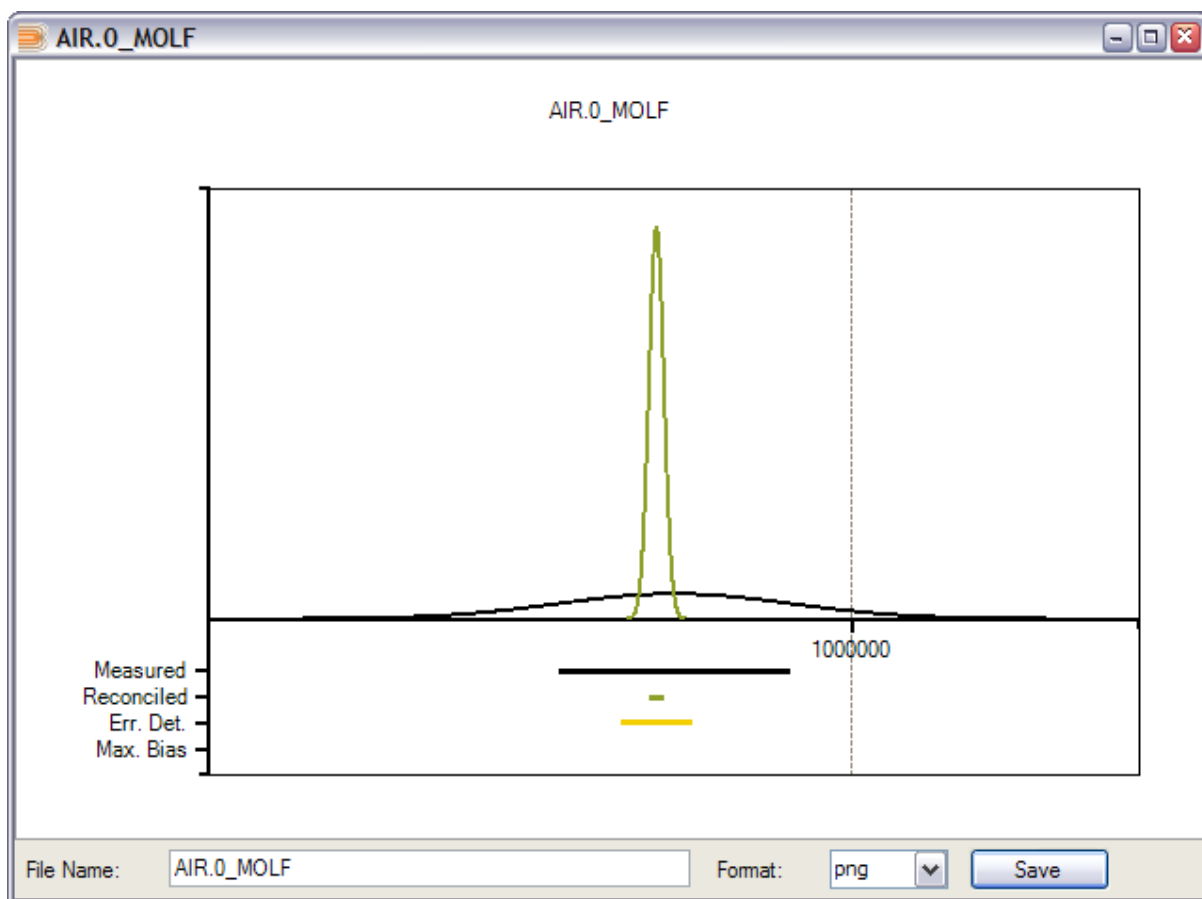


FIGURE 7 - CHART WINDOW

File name is set to tag name by default. This name can be changed using text box located at the bottom of the window. Format of output file can also be chosen between png, jpg, bmp and gif, using drop-down menu. Size of saved picture is the same as the display size. To save chart, just click the Save button. This will close the window.



FIGURE 8 - SAVE CHART PANEL

ABOUT HELP

Help file can be displayed any time, by pressing F1 key, or by clicking on the help button () in the toolbar.

SECTION 4: GLOBAL VIEW

The global view is the default view that is shown when ValiAnalytics is opened. It displays general results about your model such as list of tags, gain distribution, number of redundancies, quality of results, etc.

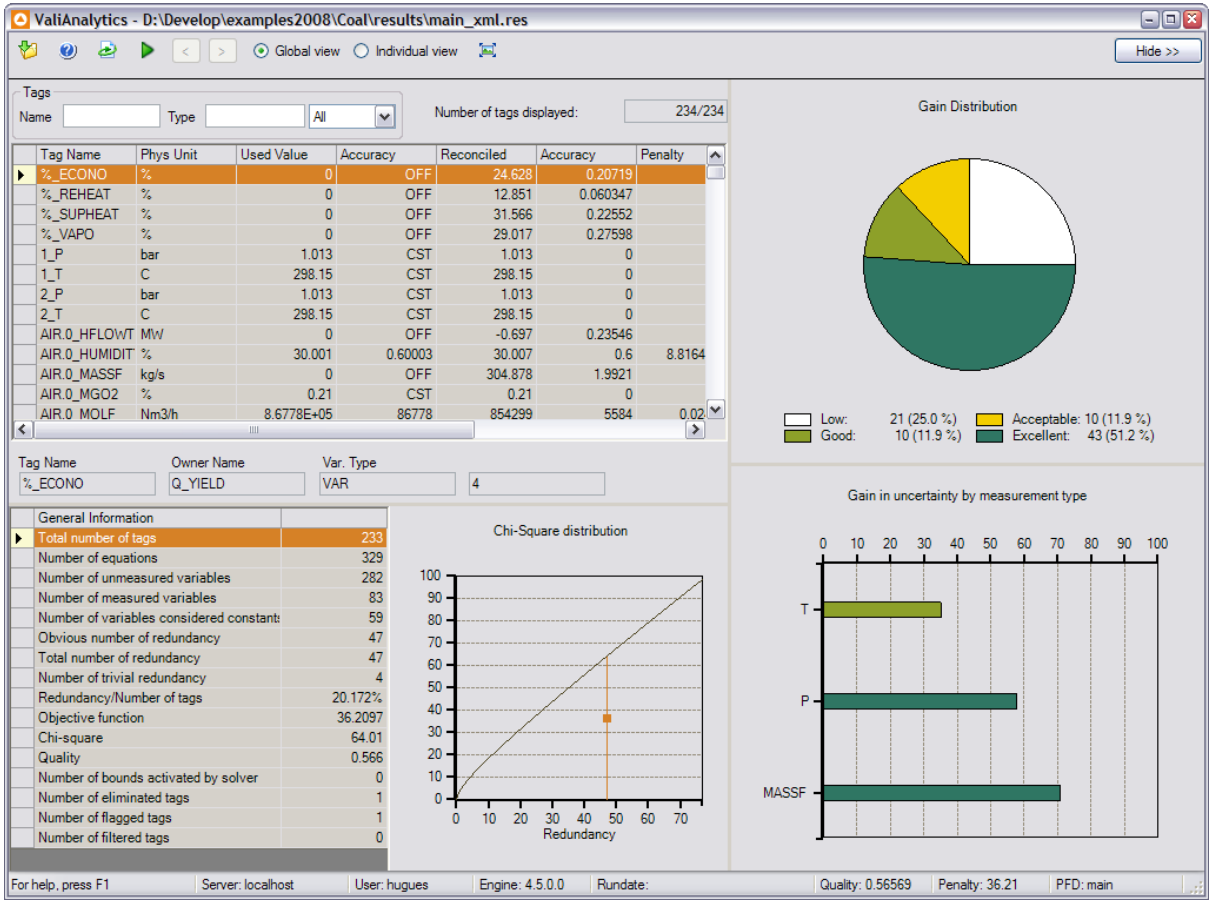


FIGURE 9 - GLOBAL VIEW

Global view is divided in four main parts:

GLOBAL TAGS TABLE

The global tags table displays the list of tags in the model, and allows filtering these tags in order to facilitate finding the tag you are looking for.

Tags							
Name		Type		All v	Number of tags displayed: 234/234		
	Tag Name	Phys Unit	Used Value	Accuracy	Reconciled	Accuracy	Penalty
▶	%_ECONO	%	0	OFF	24.628	0.20719	
	%_REHEAT	%	0	OFF	12.851	0.060347	
	%_SUPHEAT	%	0	OFF	31.566	0.22552	
	%_VAPO	%	0	OFF	29.017	0.27598	
	1_P	bar	1.013	CST	1.013	0	
	1_T	C	298.15	CST	298.15	0	
	2_P	bar	1.013	CST	1.013	0	
	2_T	C	298.15	CST	298.15	0	
	AIR.0_HFLOWT	Mw	0	OFF	-0.697	0.23546	
	AIR.0_HUMIDIT	%	30.001	0.60003	30.007	0.6	8.8164
	AIR.0_MASSF	kg/s	0	OFF	304.878	1.9921	
	AIR.0_MGO2	%	0.21	CST	0.21	0	
	AIR.0_MOLF	Nm3/h	8.6778E+05	86778	854299	5584	0.02
◀ ▶							
Tag Name		Owner Name		Var. Type			
%_ECONO		Q_YIELD		VAR		4	

FIGURE 10 - GLOBAL TAGS TABLE

GAIN DISTRIBUTION CHART

The gain distribution chart gives an overview of the number of tags for each level of gain.

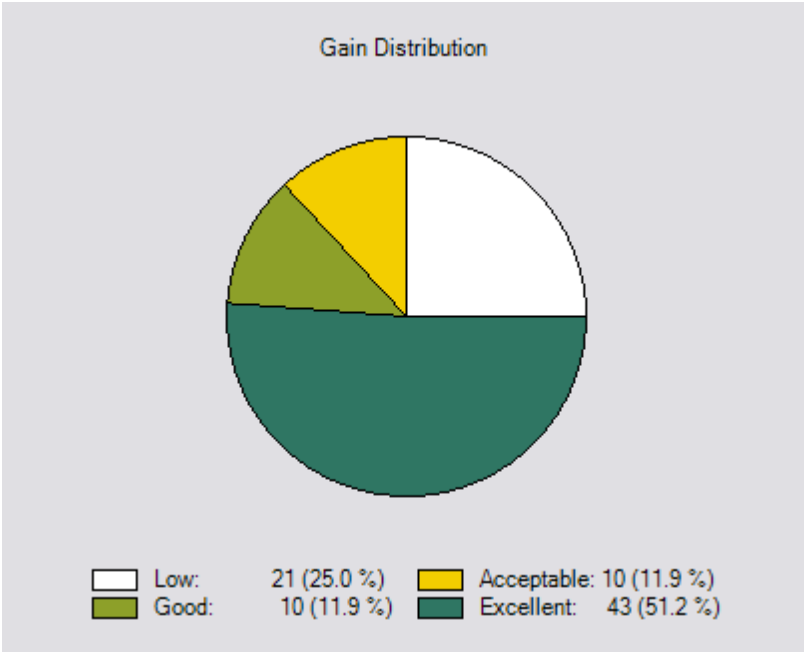


FIGURE 11 - GAIN DISTRIBUTION CHART

RUN INFORMATION

The run information part gives a general overview of reconciliation results, and a graphical representation of the chi-square curve, with actual point plotted.

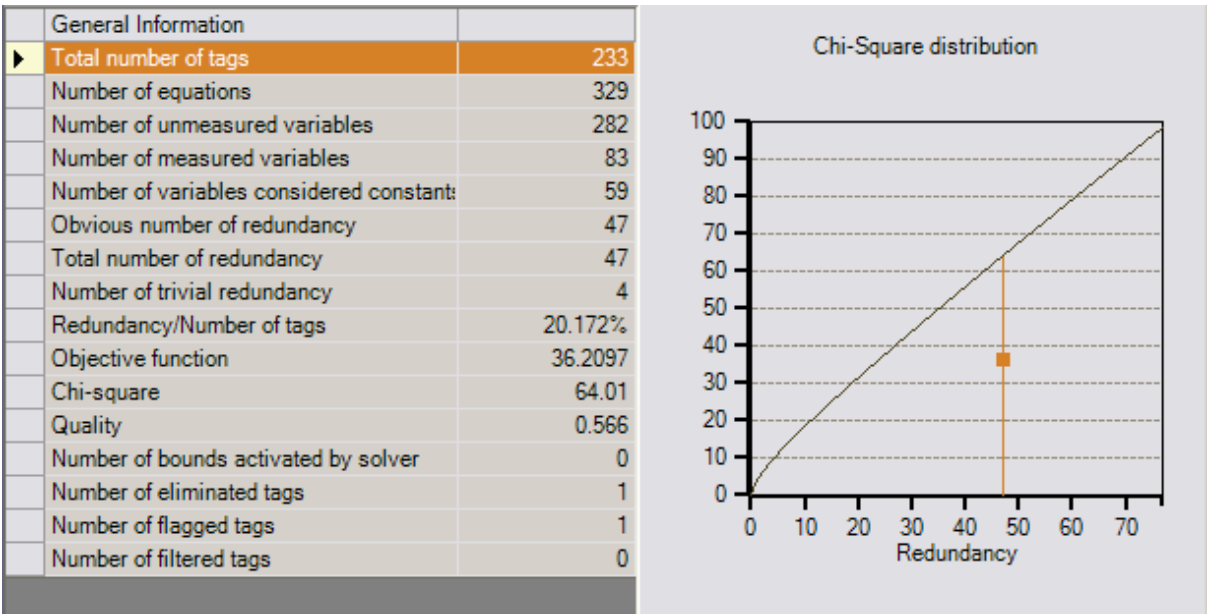


FIGURE 12 - GENERAL RUN INFORMATION

GAIN BY MEASUREMENT TYPE

The gain by measurement type chart shows the average gain for each measurement type.

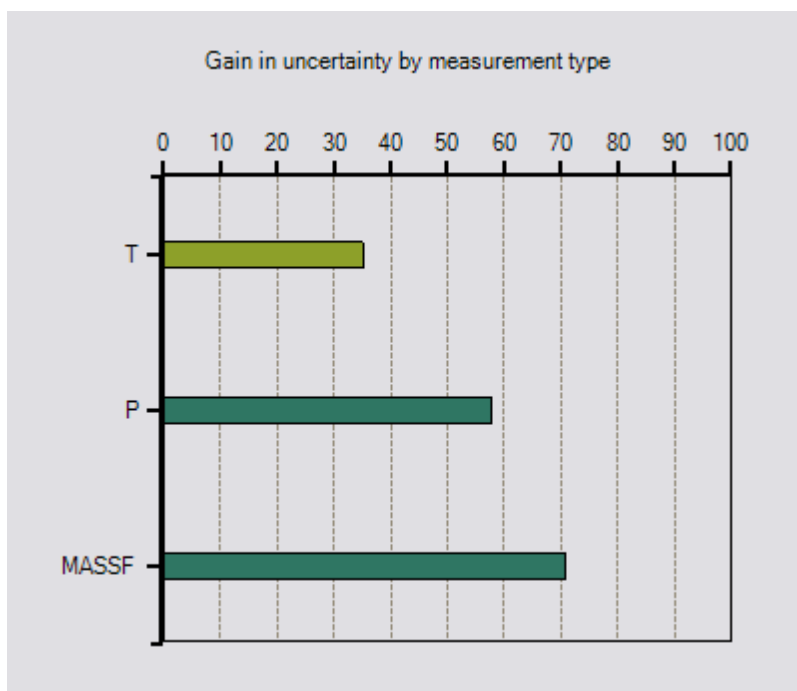


FIGURE 13 - GAIN BY MEASUREMENT TYPE

4.1 GLOBAL TAGS TABLE

The global tags table is visible in both global and individual view. It contains all information related to tags, i.e.:

- Measured value and its accuracy (in absolute): actual measured value and the accuracy for this value
- Reconciled value and accuracy (in absolute): value after reconciliation, and accuracy calculated by error propagation mechanism
- Penalty: a value that reflects the difference between the measured and reconciled values
- Gain: the gain in uncertainty of the tag
- Error detectability: the maximum error that can occur on a measurement before it is eliminated by gross error elimination mechanism
- Maximum bias: the maximum error that will occur on a reconciled value if an error occurs on the measured value, just before error detectability limit is reached for that measured value

Measured and reconciled values and accuracies, gain and penalty can also be found in ValiModeller, html reports, and are available through ValiReport. Refer to VALI help for more information on these modules.

Error detectability and maximum bias are only available in ValiAnalytics. The error detectability represents the deviation that can occur on a measured value before it is eliminated by gross error, assuming there is no error on any other measurement. The maximum bias represents the change on the reconciled value if an error equal to error detectability occurs on the measured value of this tag, i.e. just before gross error elimination.

FILTERING

At the top of the table, you can find text boxes allowing to filter the global tags table. This can be used to search on tags, but also to get information on parts of the model, as global gain charts and gain by measurement type charts are synchronized with the global tags table.



FIGURE 14 - FILTERS ON TAGS TABLE

Three different types of filtering are available:

- Name: filtering the name of tags. Enter the name of the tag(s) you are looking for. Wildcards can be used (? for one character, * for several consecutive characters)
- Type: filtering on measurement type (temperature, pressure...). Types follow VALI nomenclature (T=temperature, P=pressure, MASSF=mass flow-rate...). Wildcards can be used (? for one character, * for several consecutive characters)
- Drop-down: pre-defined criteria to select tags:
 - All: displays all tags
 - OFF: displays only tags that are OFF in the model
 - CST: displays only constants
 - Eliminated: displays only eliminated tags
 - Flagged: displays only flagged tags

At the top-right of the tags table, the number of displayed tags vs. total number of tags is given.

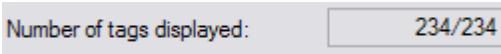


FIGURE 15 - NUMBER OF TAGS DISPLAYED

TAGS INFORMATION

At the bottom of the table, several boxes give you information about currently selected tag:

- Tag name
- Owner name: the name of the object (unit or stream) on which the tag is defined
- Var. Type: the type of variable.

Tag Name	Owner Name	Var. Type	
%_ECONO	Q_YIELD	VAR	4

FIGURE 16 - TAG INFORMATION

RIGHT CLICK

Right clicking on the table offers the possibility to export the table to MS Excel or to wordpad, and to show hidden columns, which are:

- Penalty
- Gain
- Error detectability
- Maximum bias

4.2 GAIN DISTRIBUTION CHART

The chart on the upper right part of the window gives a graphical representation of distribution of gain in accuracy on tags of the model. This chart is synchronized with tags table, which means that only tags that are displayed in the global tags table are taken into account in this chart.

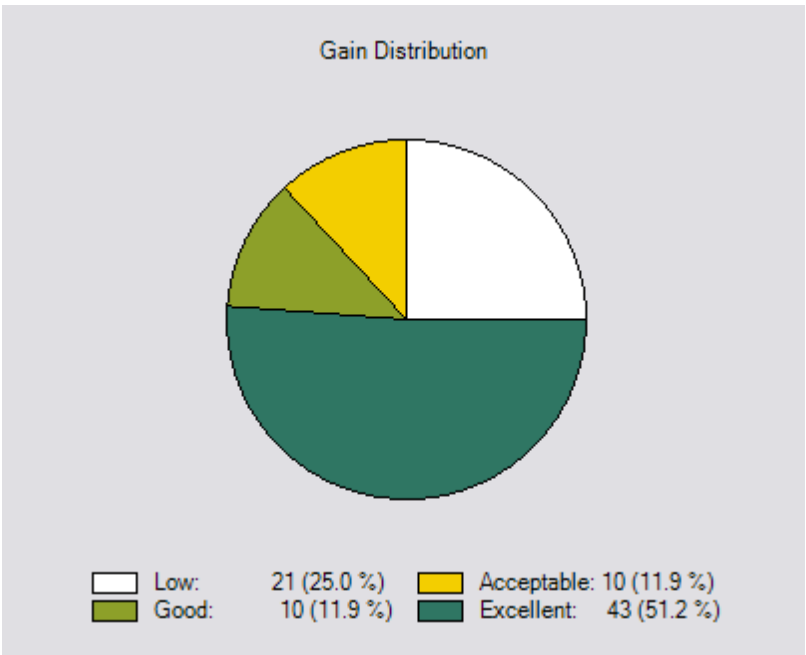


FIGURE 17 - GAIN DISTRIBUTION CHART

This pie chart is divided in four different levels:

- Low gain, white: gain is lower than 10%
- Acceptable gain, yellow: gain is between 10% and 29%
- Good gain, green: gain is between 29% and 42%
- Excellent gain, dark green: gain is above 42%

For each zone, the number of tags that have gain in this interval is given in the legend of the chart, as well as the percentage of tags that this number represents.

4.3 GENERAL INFORMATION

The lower left part of the window contains general information about current run.

On the left, a table displays general information about the model:

- Total number of tags defined in the model (can be different than number of tags used for reconciliation: streams or units OFF...)
- Number of equations in the model
- Number of variables (measured and not measured) in the model
- Number of variables considered constant
- Number of redundancy (obvious, trivial and total)
- Redundancy by number of tags (average redundancy per tag, expressed in percent)
- Objective function (sum of penalties)
- Value of chi-square test
- Quality of dataset (objective function/value of chi-square; the smaller, the better)
- Number of bounds activated by solver
- Number of eliminated tags
- Number of flagged tags
- Number of filtered tags

General Information	
▶ Total number of tags	233
Number of equations	329
Number of unmeasured variables	282
Number of measured variables	83
Number of variables considered constant	59
Obvious number of redundancy	47
Total number of redundancy	47
Number of trivial redundancy	4
Redundancy/Number of tags	20.172%
Objective function	36.2097
Chi-square	64.01
Quality	0.566
Number of bounds activated by solver	0
Number of eliminated tags	1
Number of flagged tags	1
Number of filtered tags	0

FIGURE 18 - GENERAL INFORMATION TABLE

Next to this table, a chart shows the chi-square curve, with actual value marked by an orange line and the value of the objective function is represented by an orange square. This value is below the curve when quality number is lower than one and above the curve when quality number is greater than one (the lower quality the better).

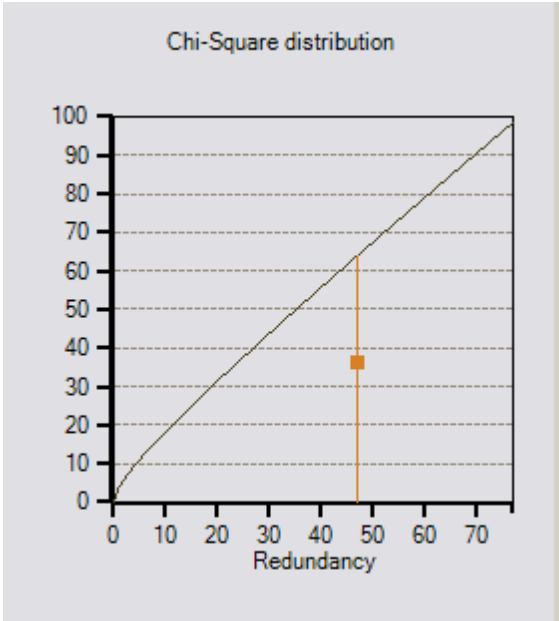


FIGURE 19 - CHI-SQUARE CHART

4.4 GAIN IN UNCERTAINTY BY MEASUREMENT TYPE

The lower right part of the window presents the average gain by measurement type. For each selected measurement type, the average gain on measurement of this type is represented. This chart is synchronized with the global tags table, which means that only tags that are displayed in the tags table are taken into account in this chart.

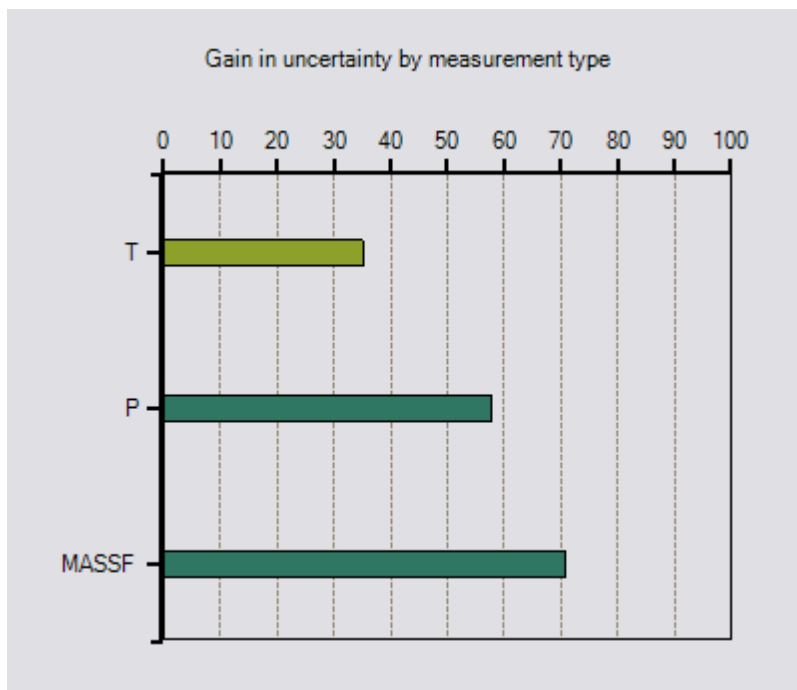


FIGURE 20 - GAIN BY MEASUREMENT TYPE

The different measurement types available in the model can be displayed by right-clicking on the chart, and selecting desired type.

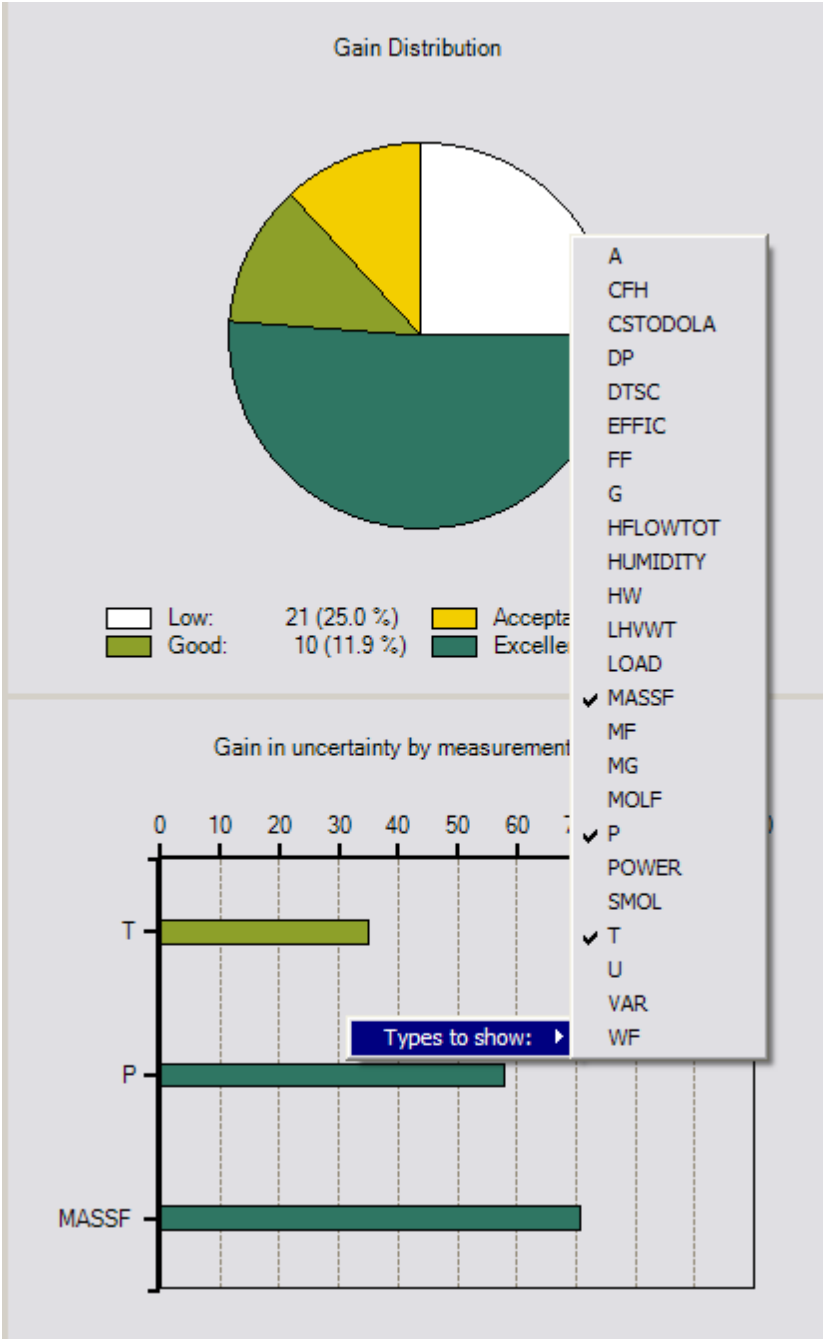


FIGURE 21 - SELECT TYPES TO SHOW

SECTION 5: INDIVIDUAL VIEW

The individual view shows results for currently selected tag. It can be shown by clicking the [Individual view button](#) located in the [toolbar](#).

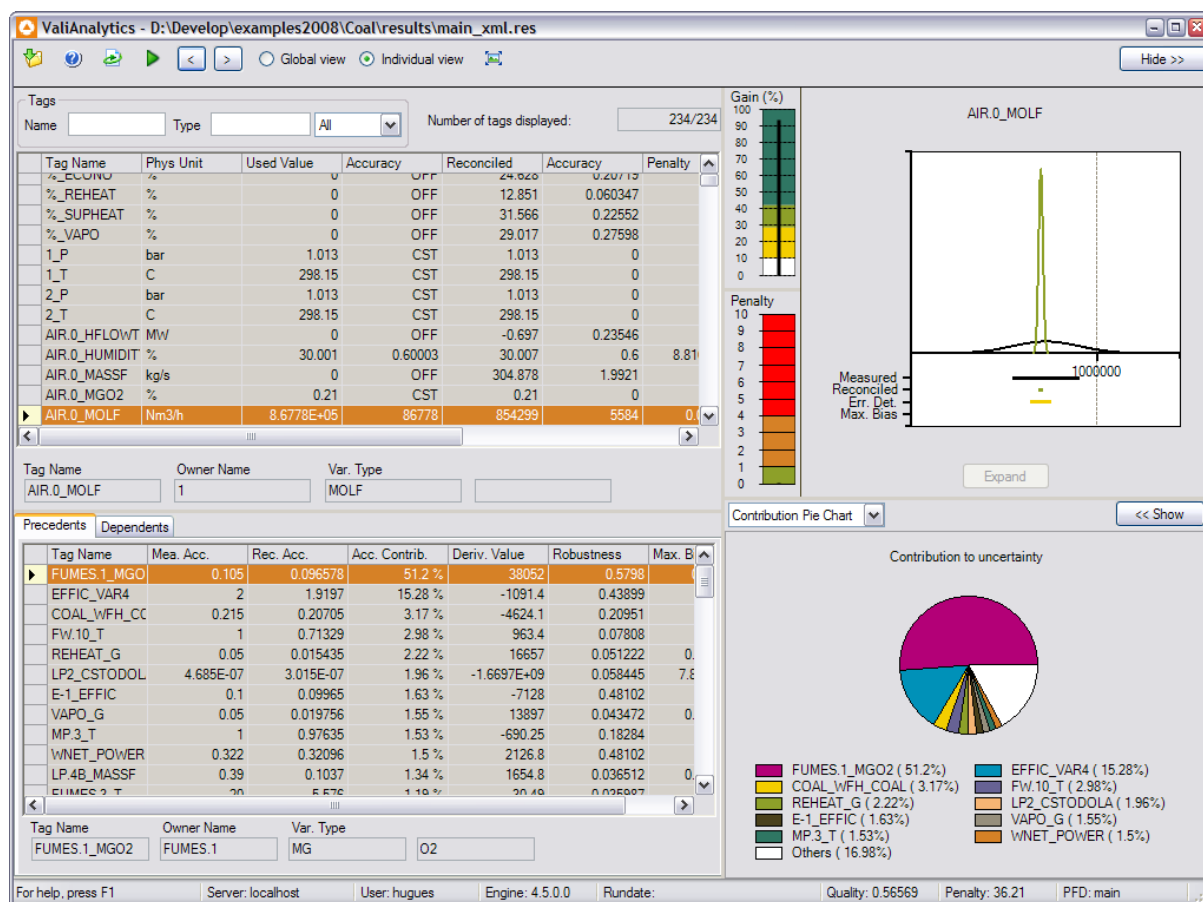


FIGURE 22 - VALIANALYTICS - INDIVIDUAL VIEW

Individual view is divided in four main parts:

GLOBAL TAGS TABLE

The global tags table displays the list of tags in the model, and allows filtering these tags in order to facilitate finding the tag you are looking for.

Tags

Name

Type

All

Number of tags displayed:

	Tag Name	Phys Unit	Used Value	Accuracy	Reconciled	Accuracy	Penalty	
▶	%_ECONO	%	0	OFF	24.628	0.20719		v
	%_REHEAT	%	0	OFF	12.851	0.060347		
	%_SUPHEAT	%	0	OFF	31.566	0.22552		
	%_VAPO	%	0	OFF	29.017	0.27598		
	1_P	bar	1.013	CST	1.013	0		
	1_T	C	298.15	CST	298.15	0		
	2_P	bar	1.013	CST	1.013	0		
	2_T	C	298.15	CST	298.15	0		
	AIR.0_HFLOWT	MW	0	OFF	-0.697	0.23546		
	AIR.0_HUMIDIT	%	30.001	0.60003	30.007	0.6	8.8164	
	AIR.0_MASSF	kg/s	0	OFF	304.878	1.9921		
	AIR.0_MGO2	%	0.21	CST	0.21	0		
	AIR.0_MOLF	Nm3/h	8.6778E+05	86778	854299	5584	0.02	v

||||

Tag Name

Owner Name

Var. Type

%_ECONO

Q_YIELD

VAR

4

FIGURE 23 - GLOBAL TAGS TABLE

TAG UNCERTAINTY DISTRIBUTION

The uncertainty distribution chart represents the Gaussian distribution of measured and reconciled values, as well as bars representing the values of gain and penalty.

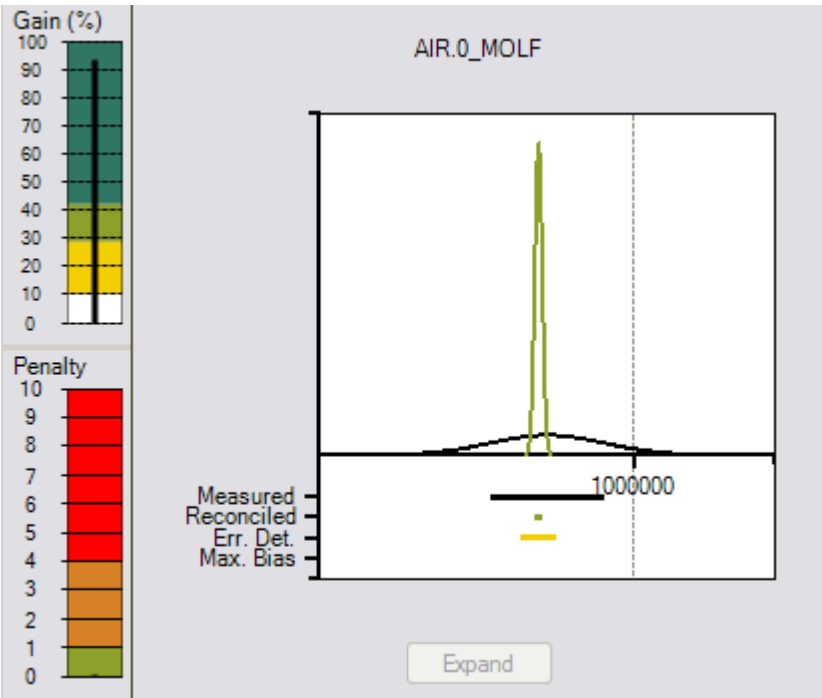


FIGURE 24 - UNCERTAINTY DISTRIBUTION

PRECEDENTS/DEPENDENTS TABLES

The precedent and dependent tables show the list of tags contributing to uncertainty of selected tag and tags of which selected tag contributes to their uncertainty.

Precedents		Dependents					
Tag Name	Mea. Acc.	Rec. Acc.	Acc. Contrib.	Deriv. Value	Robustness	Max. B	
FUMES.1_MGO	0.105	0.096578	51.2 %	38052	0.5798		
EFFIC_VAR4	2	1.9197	15.28 %	-1091.4	0.43899		
COAL_WFH_CO	0.215	0.20705	3.17 %	-4624.1	0.20951		
FW.10_T	1	0.71329	2.98 %	963.4	0.07808		
REHEAT_G	0.05	0.015435	2.22 %	16657	0.051222	0.	
LP2_CSTODOLA	4.685E-07	3.015E-07	1.96 %	-1.6697E+09	0.058445	7.8	
E-1_EFFIC	0.1	0.09965	1.63 %	-7128	0.48102		
VAPO_G	0.05	0.019756	1.55 %	13897	0.043472	0.	
MP.3_T	1	0.97635	1.53 %	-690.25	0.18284		
WNET_POWER	0.322	0.32096	1.5 %	2126.8	0.48102		
LP.4B_MASSF	0.39	0.1037	1.34 %	1654.8	0.036512	0.	
FUMES.1_T	20	5.576	1.19 %	20.49	0.025097		

FIGURE 25 - PRECEDENTS/DEPENDENTS TABLES

PRECEDENTS/DEPENDENTS CHARTS

The precedents and dependent charts show different charts related to precedent and dependent tags, that can be switched using drop-down menu.

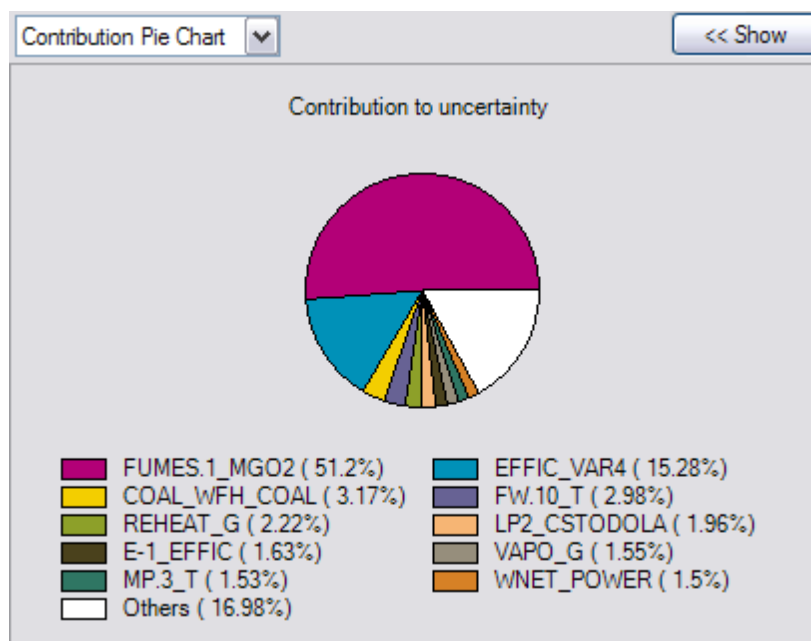


FIGURE 26 - PRECEDENTS/DEPENDENTS CHARTS

5.1 GLOBAL TAGS TABLE

The global tags table is visible in both global and individual view. It contains all information related to tags, i.e.:

- Measured value and its accuracy (in absolute): actual measured value and the accuracy for this value
- Reconciled value and accuracy (in absolute): value after reconciliation, and accuracy calculated by error propagation mechanism
- Penalty: a value that reflects the difference between the measured and reconciled values
- Gain: the gain in uncertainty of the tag
- Error detectability: the maximum error that can occur on a measurement before it is eliminated by gross error elimination mechanism
- Maximum bias: the maximum error that will occur on a reconciled value if an error occurs on the measured value, just before error detectability limit is reached for that measured value

Measured and reconciled values and accuracies, gain and penalty can also be found in ValiModeller, html reports, and are available through ValiReport. Refer to VALI help for more information on these modules.

Error detectability and maximum bias are only available in ValiAnalytics. The error detectability represents the deviation that can occur on a measured value before it is eliminated by gross error, assuming there is no error on any other measurement. The maximum bias represents the change on the reconciled value if an error equal to error detectability occurs on the measured value of this tag, i.e. just before gross error elimination.

FILTERING

At the top of the table, you can find text boxes allowing to filter the global tags table. This can be used to search on tags, but also to get information on parts of the model, as global gain charts and gain by measurement type charts are synchronized with the global tags table.

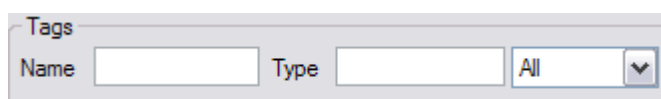


FIGURE 27 - FILTERS ON TAGS TABLE

Three different types of filtering are available:

- Name: filtering the name of tags. Enter the name of the tag(s) you are looking for. Wildcards can be used (? for one character, * for several consecutive characters)
- Type: filtering on measurement type (temperature, pressure...). Types follow VALI nomenclature (T=temperature, P=pressure, MASSF=mass flow-rate...). Wildcards can be used (? for one character, * for several consecutive characters)
- Drop-down: pre-defined criteria to select tags:
 - All: displays all tags
 - OFF: displays only tags that are OFF in the model
 - CST: displays only constants
 - Eliminated: displays only eliminated tags
 - Flagged: displays only flagged tags

At the top-right of the tags table, the number of displayed tags vs. total number of tags is given.

Number of tags displayed: 234/234

FIGURE 28 - NUMBER OF TAGS DISPLAYED

TAGS INFORMATION

At the bottom of the table, several boxes give you information about currently selected tag:

- Tag name
- Owner name: the name of the object (unit or stream) on which the tag is defined
- Var. Type: the type of variable.

Tag Name	Owner Name	Var. Type
%_ECONO	Q_YIELD	VAR

FIGURE 29 - TAG INFORMATION

RIGHT CLICK

Right clicking on the table offers the possibility to export the table to MS Excel or to wordpad, and to show hidden columns, which are:

- Penalty
- Gain
- Error detectability
- Maximum bias

5.2 TAG CHART

The upper right part of the individual view shows the Gaussian distribution of measured and reconciled values, and bars representing the value of gain and penalty of currently selected tag.

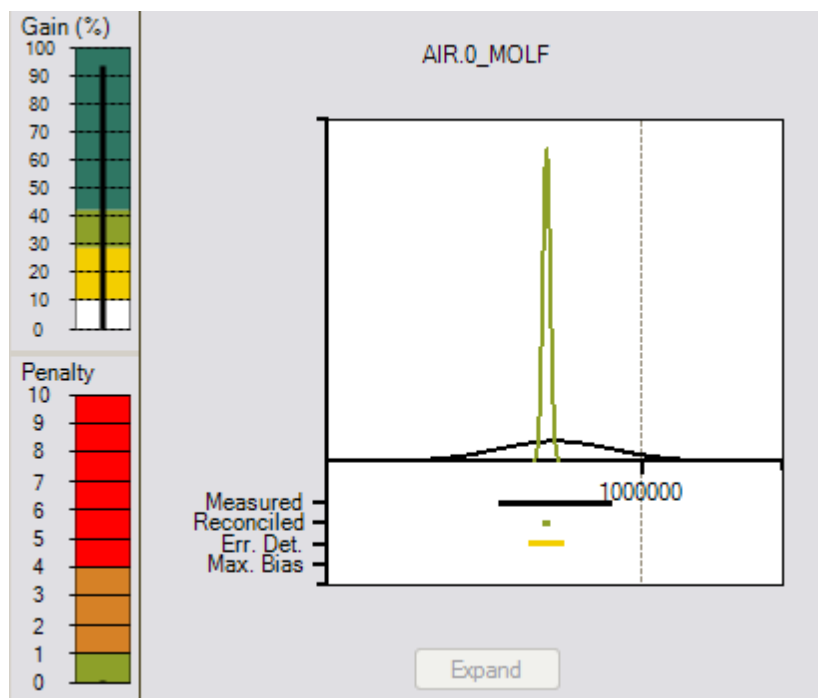


FIGURE 30 - TAG CHART

The tag chart is made of 3 different parts: the gain bar, the penalty bar and the Gaussian distribution of the tag.

GAIN BAR

The gain bar displays the gain (between 0% and 100%). The ability to reconcile the tag can be viewed directly based on color:

- White: gain lower than 10% (low gain)
- Yellow: gain between 10% and 29% (acceptable gain)
- Green: gain between 29% and 42% (good gain)
- Dark green: gain greater than 42% (excellent gain)

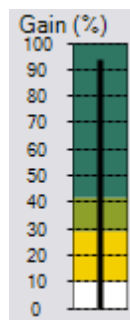


FIGURE 31 - TAG GAIN GAUGE

PENALTY BAR

The penalty bar displays the penalty. Quality of measured value can be viewed based on color:

- Green: good quality (penalty less than 1)
- Orange: medium quality (penalty between 1 and 4)
- Red: poor quality (penalty greater than 4)

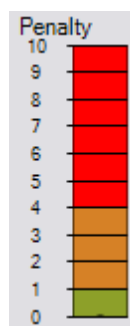


FIGURE 32 - TAG PENALTY GAUGE

TAG GAUSSIAN DISTRIBUTION

The distribution chart displays Gaussian distribution of measured and reconciled values, (upper part of the chart). It shows also a Gantt-type view of uncertainty, error detectability and maximum bias (centered on value)

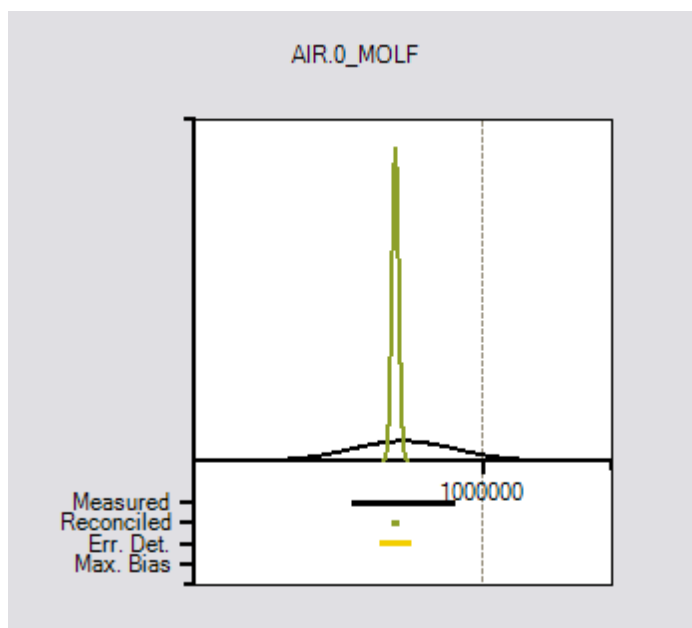


FIGURE 33 - TAG GAUSSIAN DISTRIBUTION

When error detectability is too big (i.e. greater than $100 \times$ reconciled uncertainty), the distribution's minimum and maximum axis values will be calculated according to the uncertainty of the tag, and minimum and maximum values of error detectability will not be visible. In this case, the Expand button will be enabled; by clicking on this button, minimum and maximum values of error detectability will be displayed on chart. In this case, the Gaussian distribution can be very narrow.

5.3 PRECEDENTS/DEPENDENTS TABLES

The lower left part of the individual view contains two similar pages that can be accessed using tabs:

- the precedents table, containing the list of tags that contribute to the uncertainty of currently selected tag
- the dependents table, containing the list of tags which selected tag contributes to their uncertainty

Precedents							
Tag Name	Mea. Acc.	Rec. Acc.	Acc. Contrib.	Deriv. Value	Robustness	Max. Bias	
► FUMES.1_MGO	0.105	0.096578	51.2 %	38052	0.5798	0.0	
EFFIC_VAR4	2	1.9197	15.28 %	-1091.4	0.43899		
COAL_WFH_CC	0.215	0.20705	3.17 %	-4624.1	0.20951	0	
FW.10_T	1	0.71329	2.98 %	963.4	0.07808		
REHEAT_G	0.05	0.015435	2.22 %	16657	0.051222	0.00	
LP2_CSTODOL	4.685E-07	3.015E-07	1.96 %	-1.6697E+09	0.058445	7.806	
E-1_EFFIC	0.1	0.09965	1.63 %	-7128	0.48102	0	
VAPO_G	0.05	0.019756	1.55 %	13897	0.043472	0.00	
MP.3_T	1	0.97635	1.53 %	-690.25	0.18284		
WNET_POWER	0.322	0.32096	1.5 %	2126.8	0.48102		
LP.4B_MASSF	0.39	0.1037	1.34 %	1654.8	0.036512	0.00	
FUMES.2_T	0.0	5.576	1.19 %	20.49	0.025997	0	
Tag Name	Owner Name	Var. Type					
FUMES.1_MGO2	FUMES.1	MG	O2				

Dependents							
Tag Name	Mea. Acc.	Rec. Acc.	Acc. Contrib.	Deriv. Value	Robustness	Max. Bias	
► AIR.0_MOLF	86778	5584	0.41 %	0.0041407	0.020076		
AIR.0_MASSF	1E+20	1.9921	0.41 %	1.4771E-06	0.020075		
FUMES.6_MASS	1E+20	2.0722	0.41 %	1.5336E-06	0.020037		
FUMES.1_MFO	1E+20	0.092033	0.26 %	5.3617E-08	0.015773		
O2_EXCESS	1E+20	0.55593	0.25 %	3.2249E-07	0.015706		
FUMES.1_T	1E+20	7.804	0.25 %	-4.5091E-06	0.015644		
FUMES.1_MGO	0.105	0.096578	0.25 %	5.5711E-08	0.015618	0.0	
PREHEAT_LOA	1E+20	0.8465	0.2 %	4.3625E-07	0.013953		
COAL_MW	1E+20	3.1309	0.16 %	1.4221E-06	0.012297		
HEAT_TOT_(M	1E+20	3.122	0.15 %	1.4109E-06	0.012236		
ELECTRIC_EFF	1E+20	0.15357	0.14 %	-6.733E-08	0.01187		
LP.2_MASSF	1E+20	1.000	0.12 %	4.1282E-07	0.011119		
Tag Name	Owner Name	Var. Type					
AIR.0_MOLF	1	MOLF					

FIGURE 34 - PRECEDENTS AND DEPENDENTS TABLES

These two tables contain the same information about each tag. Based on which table you are looking at, the meaning of some information can differ. Information that can be found in this table is listed below:

	PRECEDENTS TABLE	DEPENDENTS TABLE
Measured uncertainty	Uncertainty of measured value of precedent	Uncertainty of measured value of dependent
Reconciled uncertainty	Uncertainty of reconciled value of precedent	Uncertainty of reconciled value of dependent
Uncertainty contribution	Contribution of current precedent to selected tag uncertainty	Contribution of selected tag to current dependent uncertainty
Derivative value (sensitivity)	Derivative value of selected tag with respect to current precedent, i.e. how much will change selected tag if precedent is increased by 1	Derivative value of dependent tag with respect to selected tag, i.e. how much will change dependent tag if selected tag is increased by 1
Penalty	Penalty of precedent	Penalty of dependent
Robustness	Robustness of selected tag according to precedent	Robustness of dependent according to selected tag
Gain	Gain in uncertainty of precedent	Gain in uncertainty of dependent
Error detectability	Error detectability of precedent	Error detectability of dependent
Maximum bias	Maximum bias of precedent	Maximum bias of dependent
Value covariance	Value indicating how selected tag value and precedent tag value are correlated	Value indicating how selected tag value and dependent tag value are correlated
Value correlation coefficient	Value covariance weighted by square-root of product of reconciled accuracies of tags (between -100 % and 100 %)	Value covariance weighted by square-root of product of reconciled accuracies of tags (between -100 % and 100 %)
Deviation covariance	Value indicating how selected error on selected tag value and precedent tag value are correlated	Value indicating how selected error on selected tag value and dependent tag value are correlated

	PRECEDENTS TABLE	DEPENDENTS TABLE
Deviation correlation coefficient	Deviation covariance weighted by square-root of product of reconciled accuracies of tags (between -100 % and 100 %)	Deviation covariance weighted by square-root of product of reconciled accuracies of tags (between -100 % and 100 %)

Refer to [tags table](#) to get more information about error detectability and maximum bias.

TAGS INFORMATION

At the bottom of the table, several boxes give you information about currently selected tag:

- Tag name
- Owner name: the name of the object (unit or stream) on which the tag is defined
- Var. Type: the type of variable.

Tag Name	Owner Name	Var. Type	
FUMES.1_MGO2	FUMES.1	MG	O2

FIGURE 35 - TAG INFORMATION

This information is similar to the one in the general tags table.

INTERACTION

Right clicking on the table offers the possibility to export the table to MS Excel or to wordpad, and to show hidden columns, which are:

- Gain
- Penalty
- Maximum bias
- Error detectability
- Robustness
- Value covariance
- Value correlation coefficient
- Uncertainty covariance
- Uncertainty correlation coefficient

Precedents		Dependents					
Tag Name	Mea. Acc.	Rec. Acc.	Acc. Contrib.	Deriv. Value	Penalty	Robustne	
► FUMES.1_MGO	0.105	0.096578	51.2 %	38052	0.1643		
EFFIC_VAR4	2	1.9197	15.28 %	-1091.4	0.10578	0	
COAL_WFH_CC	0.215	0.20705	3.17 %	-4624.1	0.011966	0	
FW.10_T	1	0.71329	2.98 %	963.4	1.2932	0	
REHEAT_G	0.05	0.015435					
LP2_CSTODOL	4.685E-07	3.015E-07					
E-1_EFFIC	0.1	0.09965					
VAPO_G	0.05	0.019756					
MP.3_T	1	0.97635					
WNET_POWER	0.322	0.32096					
LP.4B_MASSF	0.39	0.1037					
FUMES.2_T	20	5.576	1.10 %	20			

Tag Name	Owner Name	Var. Type	
FUMES.1_MGO2	FUMES.1	MG	O2

FIGURE 36 - PRECEDENTS TABLE - OPTIONAL COLUMNS

By right clicking and selecting "Filter Values...", you also get the possibility to set some filters on values displayed, i.e. to show only tags that have accuracy contribution, derivative value and robustness in a chosen interval.

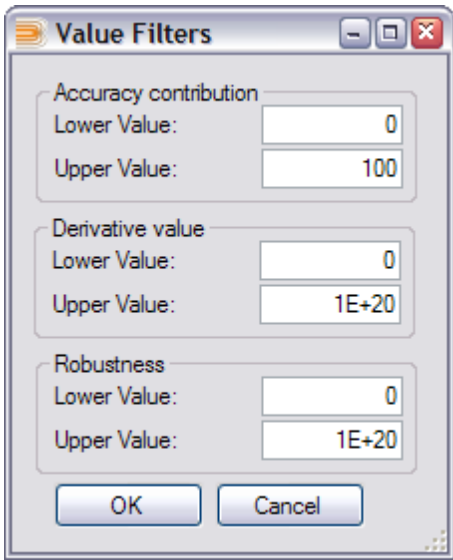


FIGURE 37 - FILTER VALUES WINDOW

Double clicking on a tag will select this tag in the general tags table. If tag is hidden due to filtering, a window will propose you to remove filtering in order to show requested tag.

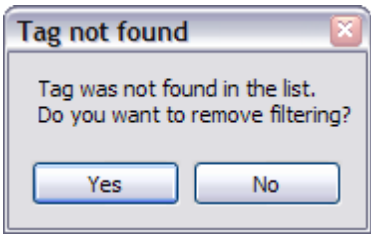


FIGURE 38 - REMOVE FILTERING WINDOW

5.4 PRECEDENTS CHARTS

The lower right part of the individual view of ValiAnalytics displays several charts related to precedent tags.

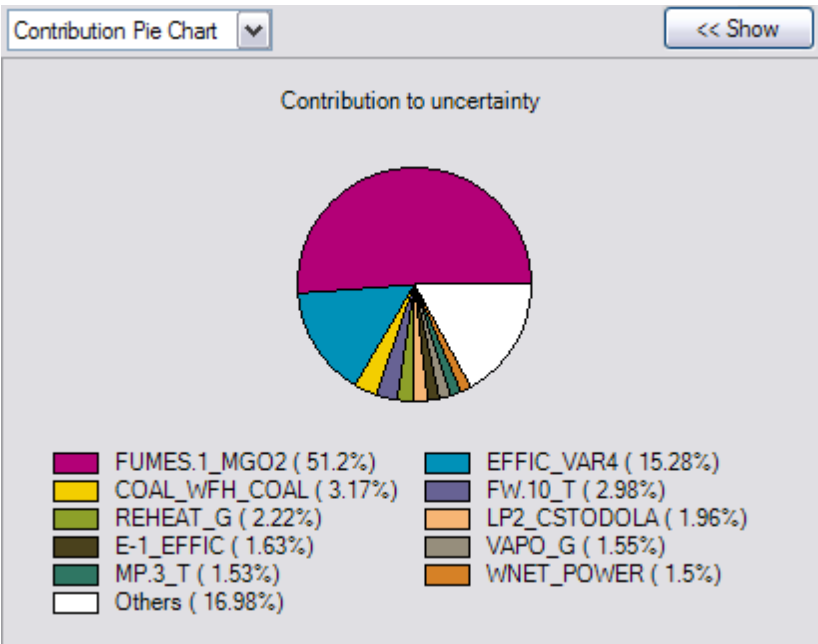


FIGURE 39 - PRECEDENTS CHARTS

DROP DOWN LIST

The different charts related to precedents and dependents are selected using a drop-down menu located in the upper left corner of the chart. Description of the represented information can be found in the [concepts table](#).

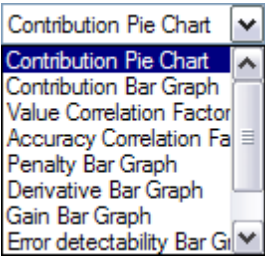


FIGURE 40 - PRECEDENTS CHARTS DROP DOWN

The different charts show information related to up to 10 precedents, corresponding to the 10 main contributors to uncertainty of the selected tag. If the selected tag has more than 10 precedents, these will be gathered under the term "Others". 9 different charts can be shown:

- Contribution Pie chart: Pie chart representing the partition of contribution of precedent tags to uncertainty of currently selected tag. Contribution to uncertainty, in percentages, of each precedent is given in the legend of the chart.

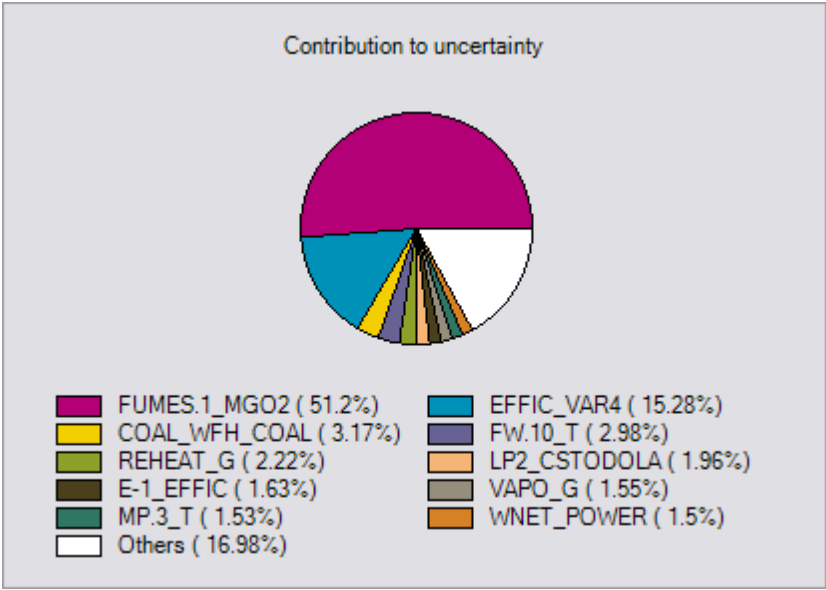


FIGURE 41 - PRECEDENTS CONTRIBUTION PIE CHART

- Contribution bar graph: Bar graph representing the contribution of precedent tags to uncertainty of currently selected tag.

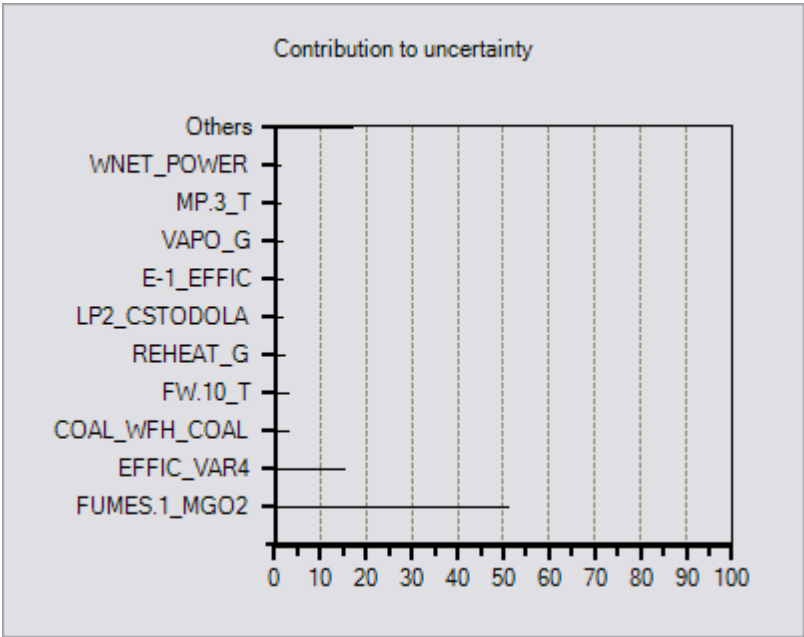


FIGURE 42 - PRECEDENTS CONTRIBUTION BAR GRAPH

- Value correlation coefficient: Bar graph representing Value correlation coefficient of main contributors.

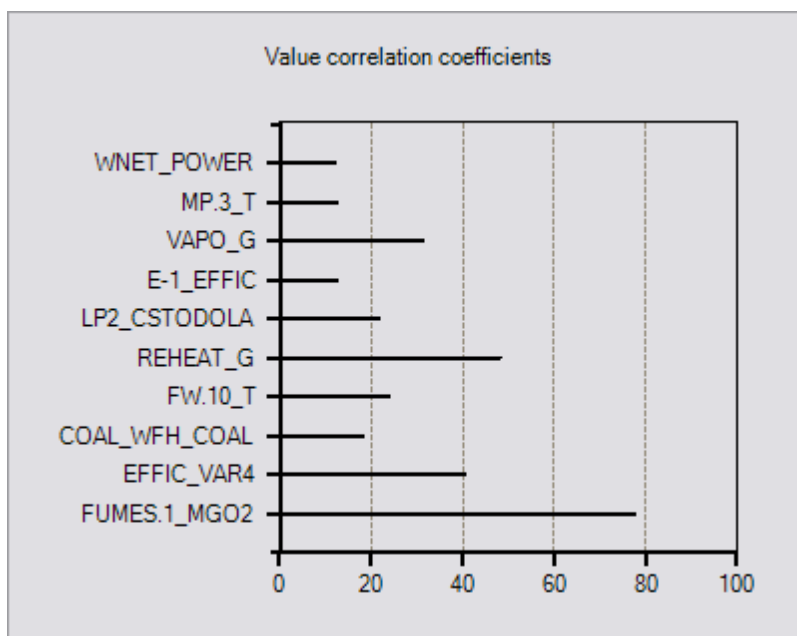


FIGURE 43 - PRECEDENTS VALUE CORRELATION COEFFICIENT BAR GRAPH

- Accuracy correlation coefficient: Bar graph representing correlation coefficient between selected tag reconciled value and precedent accuracy

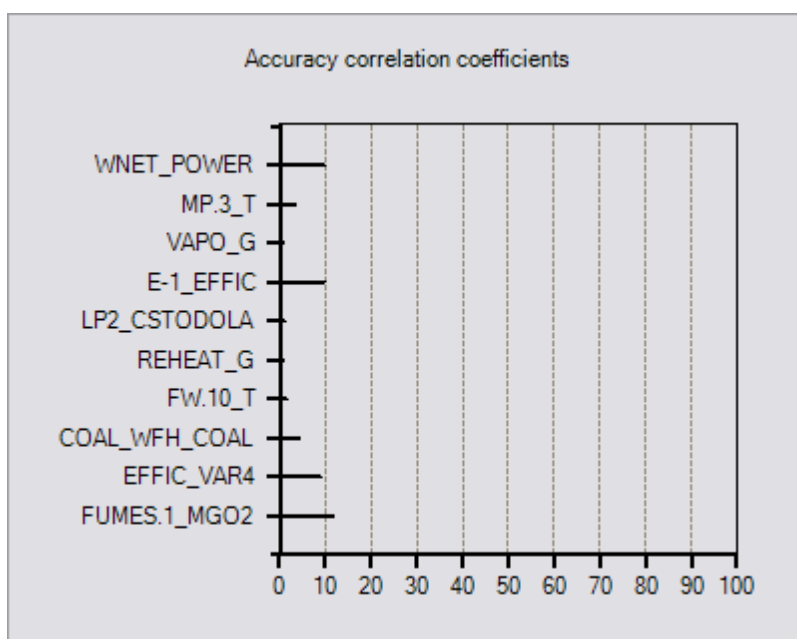


FIGURE 44 - PRECEDENTS ACCURACY CORRELATION COEFFICIENT BAR GRAPH

- Penalty bar graph: Bar graph representing the penalty of main contributors.

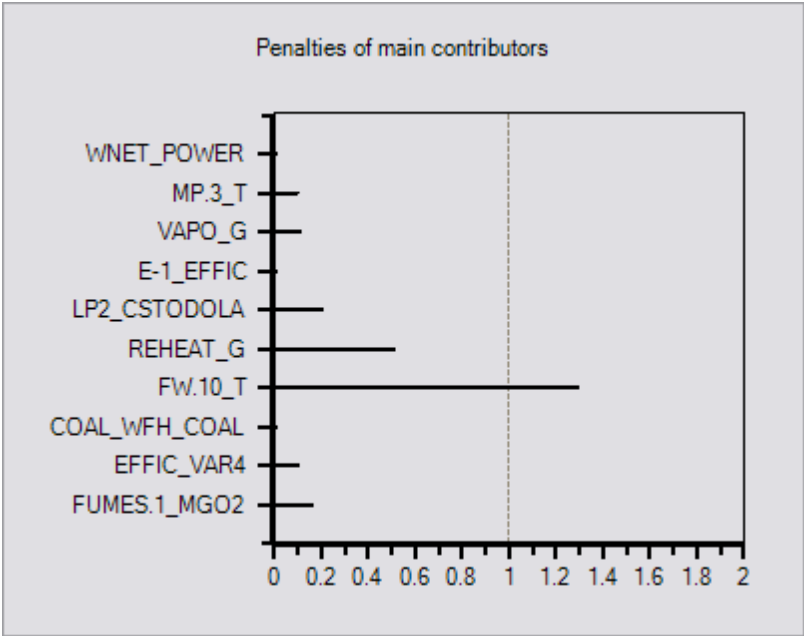


FIGURE 45 - PRECEDENTS PENALTIES BAR GRAPH

- Derivative value bar graph: Bar graph representing derivative values of selected tag with respect to precedents

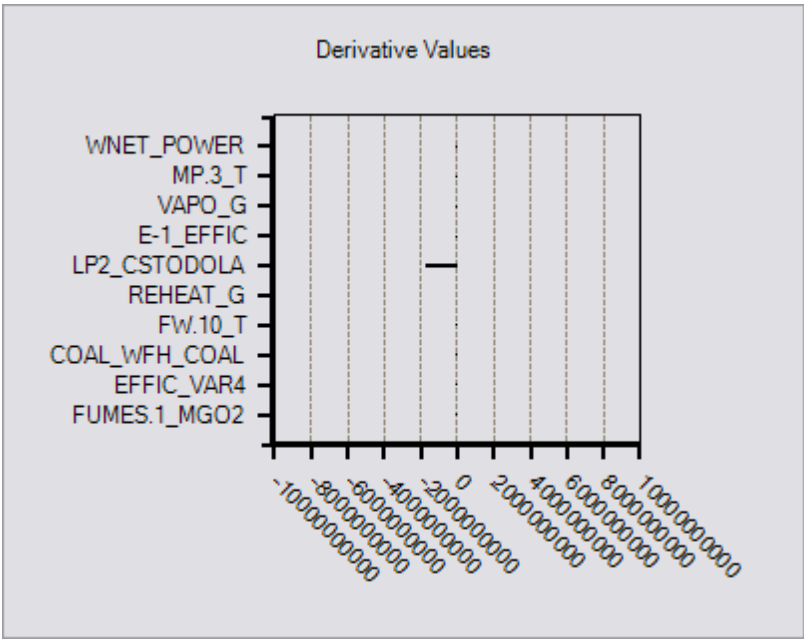


FIGURE 46 - PRECEDENTS PENALTIES BAR GRAPH

- Gain bar graph: Bar graph representing the gain in uncertainty of main contributors

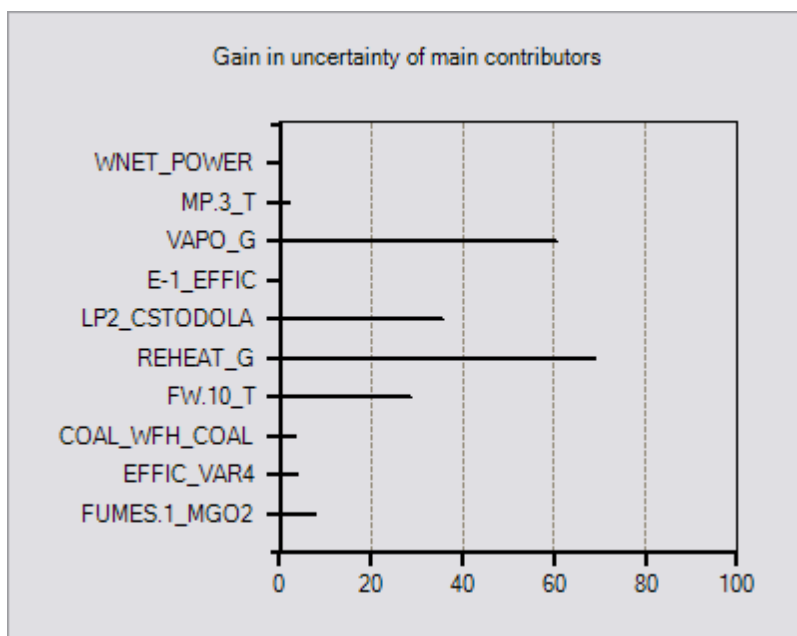


FIGURE 47 - PRECEDENTS GAINS BAR GRAPH

- Error detectability bar graph: Bar graph representing error detectability of main contributors

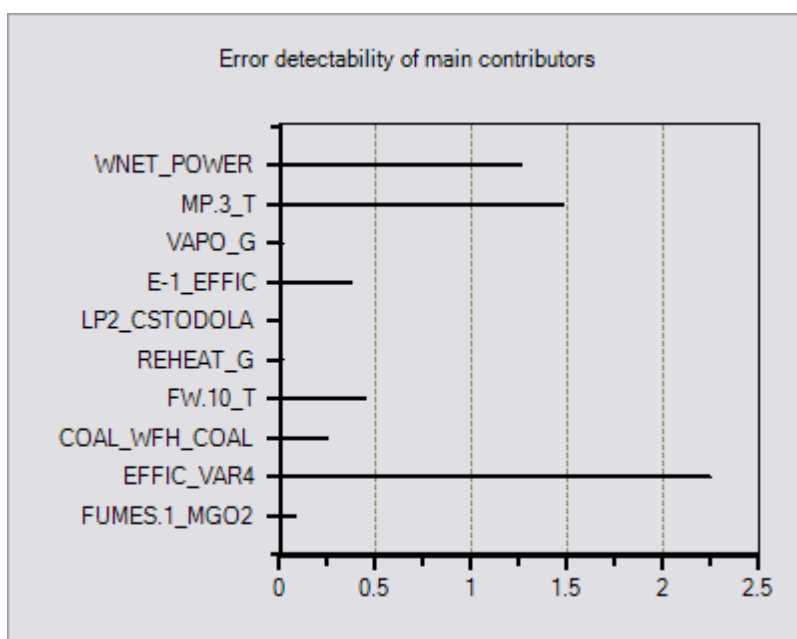


FIGURE 48 - PRECEDENTS ERROR DETECTABILITIES BAR GRAPH

- Max bias bar graph: Bar graph representing maximum bias of main contributors

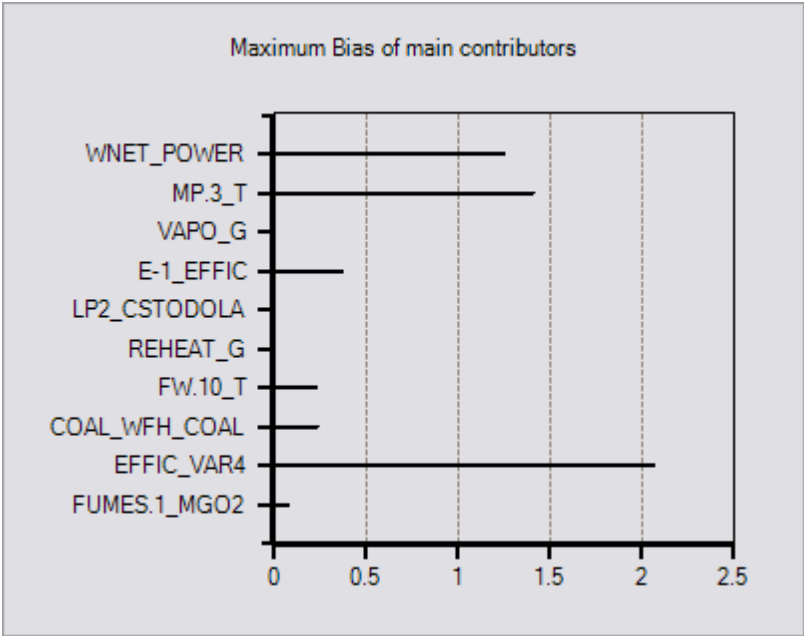


FIGURE 49 - PRECEDENTS MAXIMUM BIASES BAR GRAPH

- Robustness bar graph: Bar graph representing robustness of selected tag according to main contributors

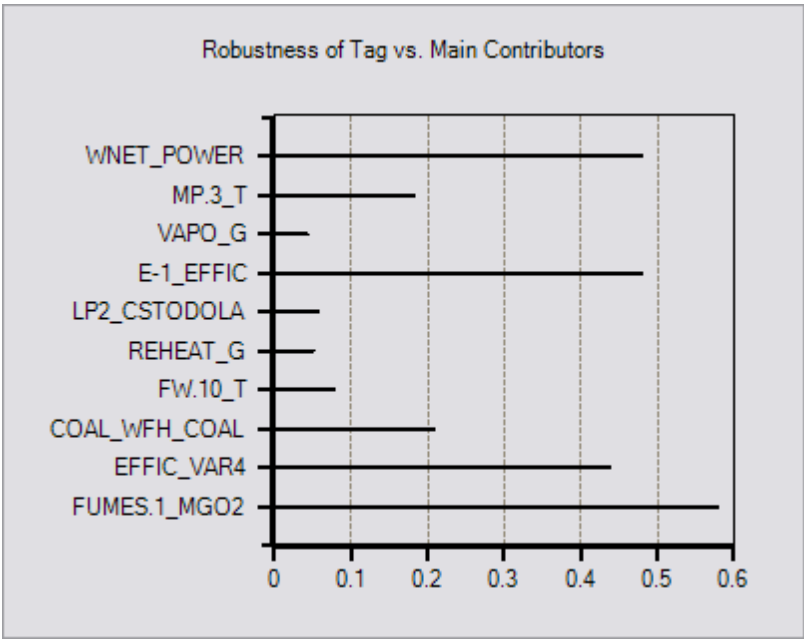


FIGURE 50 - ROBUSTNESS OF SELECTED TAG ACCORDING TO PRECEDENTS BAR GRAPH

A definition of all the concepts shown in these charts can be found in the [table of description of parameters](#).

<< Show

Hide >>

SHOW/HIDE BUTTON

When opening ValiAnalytics individual view, one chart is displayed. By clicking the Show button located at the upper right of the chart, a second chart, similar to first will be displayed. These two parts can be shown independently to show two different charts next to each other. Once the second chart is displayed, the button is renamed Hide. Clicking this button hides the second chart.



belsim

CHAPTER 11

LICENSEMANAGER

SECTION 1: INTRODUCTION

The use of Belsim software is controlled by the Belsim LicenseServer, which controls:

- the access of the user to the various modules of Belsim software according to the licenses acquired by the customer and to the rights granted to the users
- the list of plants with their links to external systems
- the list of tags and their corresponding name in external systems (PIMS, DCS, etc.)
- the list of ValiProduction and ValiPerformance applications processed by ValiService

Several LicenseServers can be available on a given network. The server specified in the belsim.ini file is selected by default.

To configure the LicenseServer and maintain the list of users, plants and tags, you will use the [LicenseManager](#). It also enables to release licenses held by applications and to make back-up of the database of registered tags.

SECTION 2: BELSIM LICENSE SERVER

2.1 LICENSING PHILOSOPHY

Different licensing policies are applied on the various modules of Belsim software family. They combine the notion of:

- instance: running module at a given moment
- simultaneous users : the total number of instances of a given module at a given moment cannot exceed the licensed number of "simultaneous users"
- tags: the total number of tags used in all models that are maintained cannot exceed the licensed number of tags
- ValiProduction Application: the set of VALI modules (shared or not shared) required to compute and report on the mass or material balance of a given production facility for the production accounting of the said facility

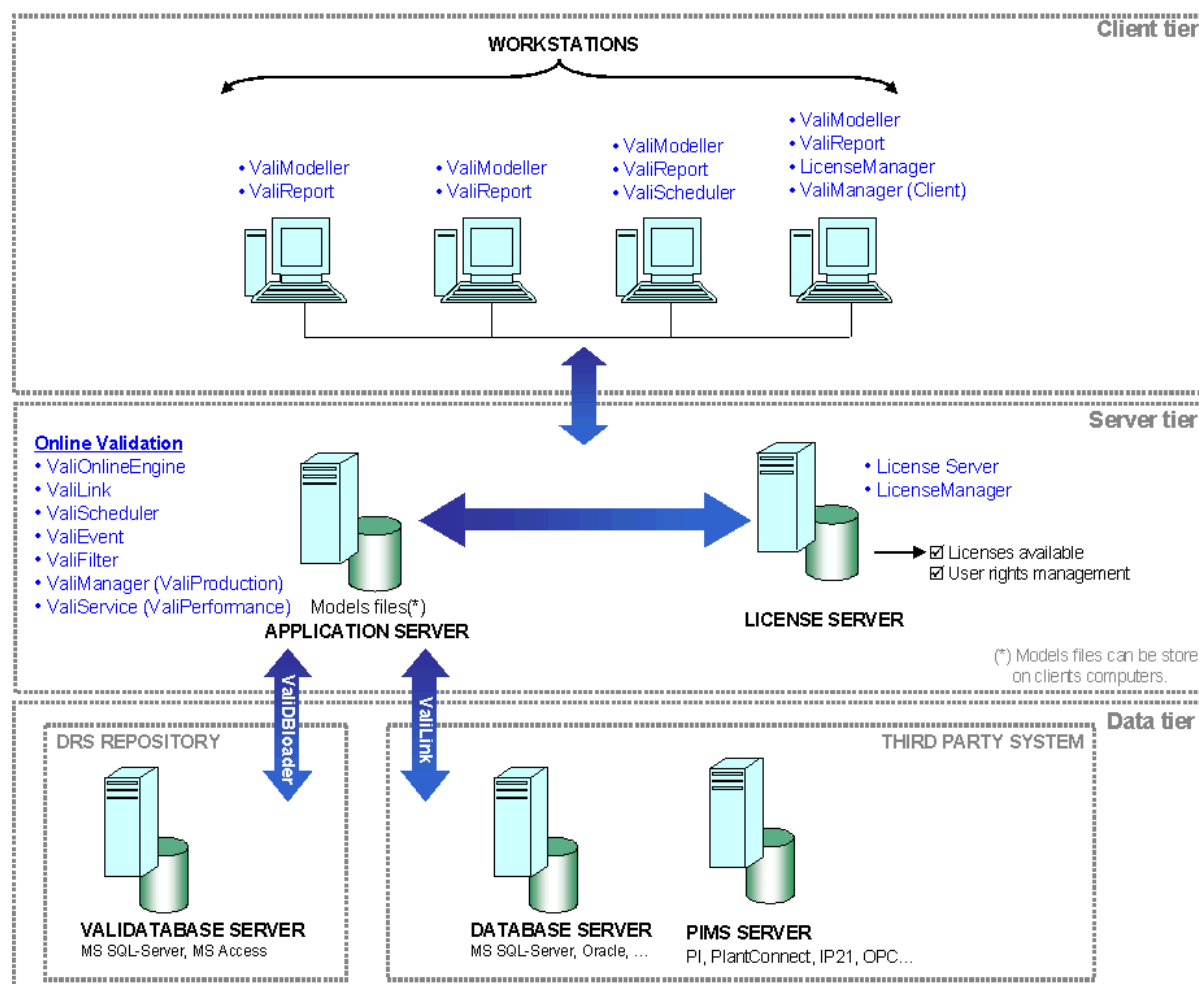


FIGURE 1 - ARCHITECTURE OF A MULTI TIER VALI INTEGRATION

2.2 USERS RIGHTS

Any application requesting a license from the LicenseServer has to provide a user name, which by default is the login name of the user in the operating system. Should this user not been defined in the LicenseServer he will be granted the rights associated to the special user GUEST. These rights concern:

- the access to the available software licenses
- the rights to create, update and delete plants and tags
- the number of tags that can be owned by the user
- the access authentication to a ValiProduction application with a complete interaction with Windows Active Directory

User rights are defined through the LicenseManager by administrators. The status of administrator can be granted to any user by any administrator.

Existing users are by default Active. They can be set as Disabled by an administrator. A Disabled user cannot perform any action.

2.2.1 The user ADMIN and Administrators

The user ADMIN is predefined and cannot be deleted. It has the status of administrator and can therefore create, update and delete users from the LicenseServer.

In addition to the rights devoted to all administrators, the user ADMIN is the only user who can change his own rights as well as the rights of the user GUEST.

The default password of the ADMIN user is "admin". This password should be changed after the installation of LicenseServer.

Attention: The notion of administrator is completely independent from the administrator status of Windows operating system.

2.2.2 The user GUEST

The user GUEST is predefined and cannot be deleted. Its rights will be allocated to any user that would access the LicenseServer without having been defined. For example, GUEST may be allowed to start a ValiModeller application but not to create any Tag or Plant.

2.2.3 The user SYSTEM

The user SYSTEM is predefined and cannot be deleted. Its rights will be used to execute ValiService applications (ValiProduction and ValiPerformance). It should have the rights to execute all the VALI modules (ValiOnlineEngine, ValiLink, ValiDBLoader, etc.)

2.2.4 The user SUBLIC

The user SUBLIC is predefined and cannot be deleted. It will be the owner of the tags imported from a sub-LicenseServer. Those tags will not be saved and will not count as registered tags of the current LicenseServer. This user is only used for internal purpose.

2.3 HANDLERS AND TIME-OUT

Each time a user starts an application, the availability of the corresponding license will be checked. If a license is available, a handler is allocated to the application. The handler is released when quitting the corresponding application.

If for any reason, the application would be abruptly terminated, its handler will be de-activated after a configurable time-out (which is by default 15 minutes but can be increased via the LicenseManager). This de-activation frees up the corresponding license.

Handlers can also be de-activated manually by an administrator through the LicenseManager.

2.4 TAGS

The number of tags you can handle with a given license server is limited in two ways:

- at any time, the total number of registered tags cannot exceed the number of tags you have been licensed
- in a given month you cannot import more tags than the number of tags you have been licensed

2.4.1 Creating and deleting tags

Each time you create tags in your models (using ValiModeller) they will be registered into the LicenseServer at the time you save your model. When you delete tags from your models, they will not be deleted from the LicenseServer. Tags can only be removed from the LicenseServer via the LicenseManager.

2.4.2 Importing tags

Tags can also be imported:

- using ValiModeller when loading a model which contains tags that are not yet registered
- using the LicenseManager, when importing a list of tags (and possibly their DBname).
Note that only the new tags will be considered as imported. The already existing tags will simply be updated

The number of imported tags is reset at the beginning of each month.

In case of emergency, you can reset the number of imported tags to zero by contacting Belsim hotline support.

2.5 PLANTS

Each tag of a model must be associated to a PLANT. This enables to easily track the tags at the License Server level.

The notion of PLANT is also used by ValiLink to communicate with external systems. For each plant, you will indicate the type of system (for example PI or MS-SQL) that gives access to this tag as well as the IP address (or UNC) and the username and password to be used for the connection.

2.6 ValiManager and ValiPerformance

Definition of applications processed by ValiService is stored in the LicenseServer.

ValiManager is the set of VALI modules (shared or not shared) required to compute and report on the mass or material balance of a given production facility for the production accounting of the said facility. ValiManager applications are configured through ValiManager client application.

ValiPerformance is the set of VALI modules required to compute and report on a performance monitoring application. ValiPerformance applications are configured through ValiScheduler application.

2.7 SUB-LICENSES

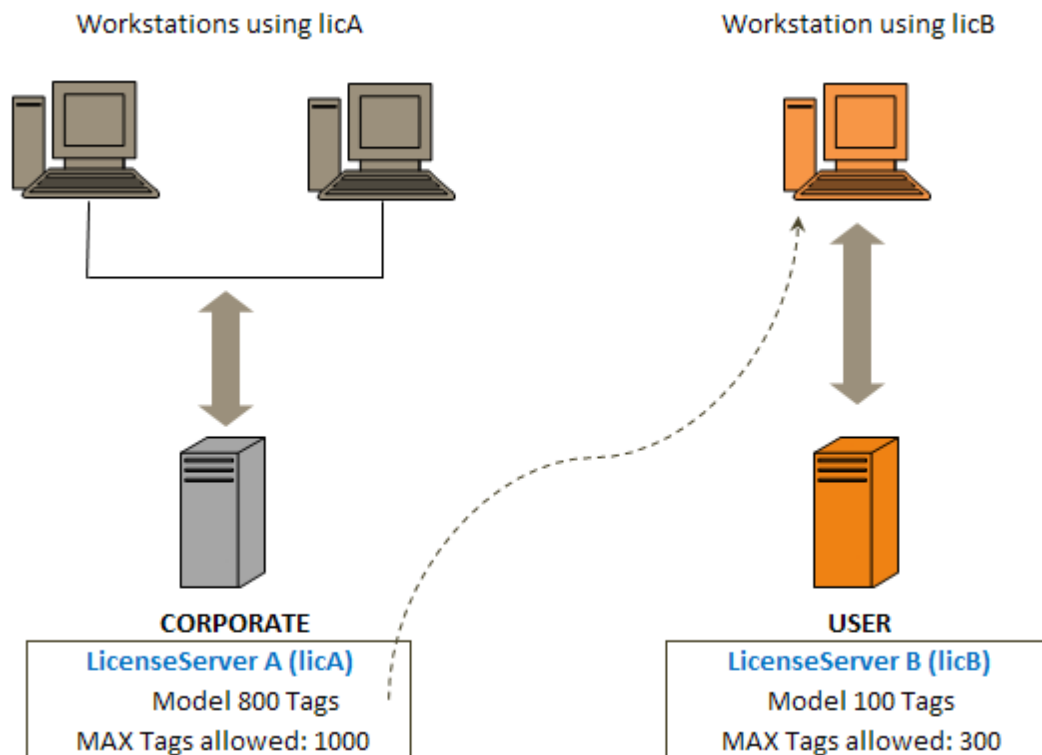
The Sub-Licensing features allow the USER to work on models created on other LicenceServers (e.g. Corporate Servers), with the local LicenceServer installed on the USER own laptop or workstation. Additional Tags and Plants registration or importation on the local LicenseServer are not required.

The present section aims at describing the Sub-Licensing feature. A typical situation is given as an example in order to illustrate the Sub-Licensing concept.

2.7.1 THE SITUATION

As illustrated below:

- the Corporate LicenseServer (licA) owns 800 Tags out of 1000 allowed
- the USER Workstation LicenseServer (licB) owns 100 Tags out of 300 allowed



The licB USER working on his own PC (orange colored), would like to work on the model owning 800 Tags that has been created on licA.

As the USER LicenseServer (licB) maximum Tags allowed is 300, and as 100 Tags have already been registered, the USER is only allowed to register 200 more Tags. The 800 Tags model can thus not be imported on the USER workstation and the tags cannot be registered as the number of tags to be registered exceeds the 200 tags free.

2.7.2 THE SUB-LICENSING

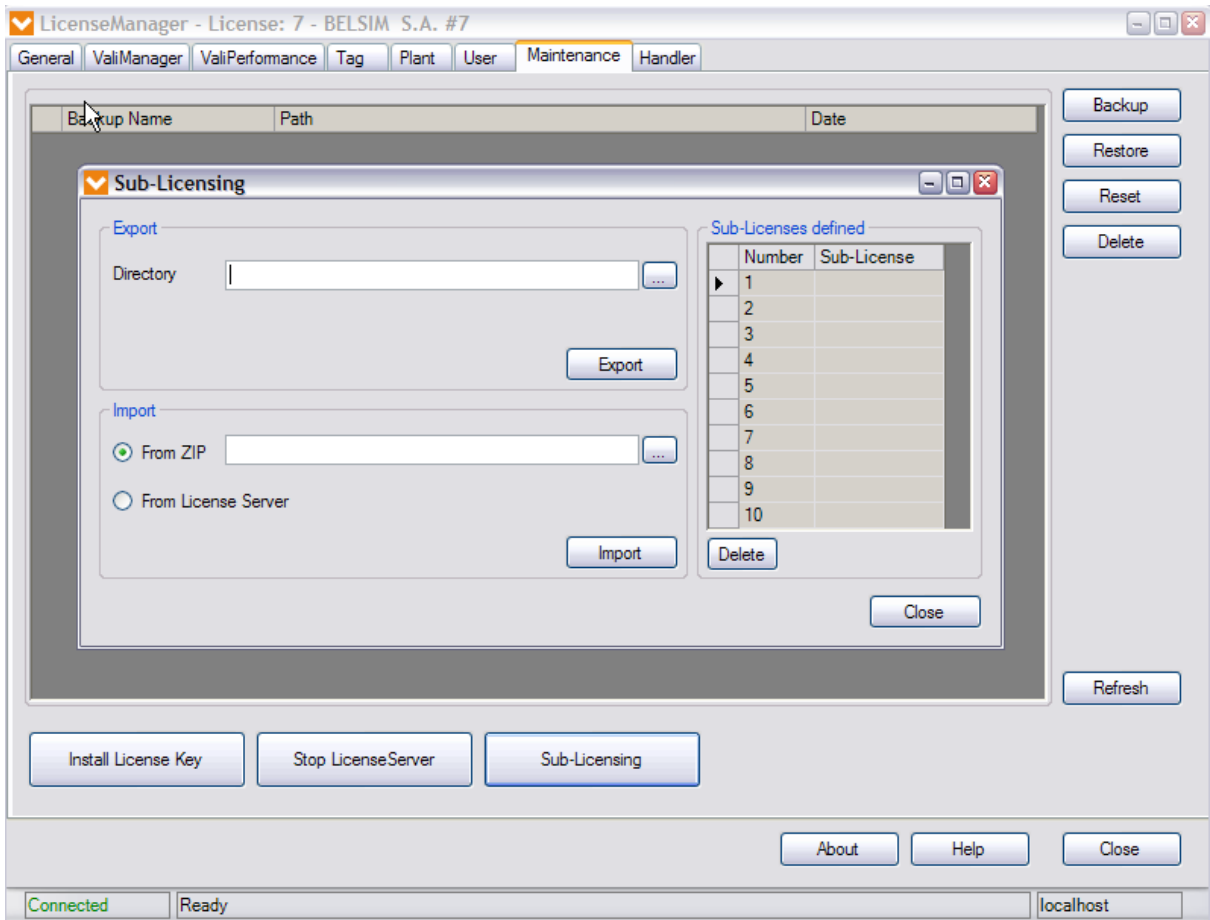
To handle the situation, the sub-licensing procedure is implemented which allows, for a given period of time, to copy and import the database from licA to licB:

- the USER working on his own workstation using licB, is allowed to share the tags defined on licA while licB still have the 200 tags free
- the USER working on his own workstation using licB, is allowed to work and modify the imported model. The USER can create up to 200 new tags in the imported model

2.7.3 THE PROCEDURE

EXPORT DATABASE FROM LICA

- start the LicenseManager and connect on LicenseServer A using “Admin” user
- go to the Maintenance Tab
- click on the Sub-Licensing button
- under Export section, define the location where you want to save the ZIP file
- click the Export button



IMPORT DATABASE FROM LICB

- start the LicenseManager and connect on LicenseServer B using “Admin” user
- go to the Maintenance Tab
- click on Sub-Licensing button
- under Import section, select “From Zip” option and locate the ZIP file previously exported
- click Import button

IMPORT USING A CONNECTION BETWEEN LICA AND LICB

- start the LicenseManager and connect on LicenseServer B
- go to the Maintenance Tab
- click on Sub-Licensing button
- under Import section, select “From License Server” option
- click Import button
- you will be prompted to log on LicenseServer A using “Admin” user
- click OK

2.7.4 DELETING SUB-LICENSES

- start the LicenseManager and connect using “Admin” user
- go to the Maintenance Tab
- click on Sub-Licensing button
- select the Sub-Licenses to be deleted and click the Delete button

2.7.5 GENERAL INFORMATION

- the TAGs/Plants defined in a Sub-License belong to a user called SUBLIC. These TAGs/Plants cannot be modified or deleted
- you must be logged as ADMIN to perform Sub-Licensing import/export/delete
- it is not required to launch LicenseManager on the computer where the LicenseServer is installed to export or import the database. You can do it from your workstation. The database will be transferred through the network connection
- you can define up to 10 Sub-Licenses on the local LicenseServer. When you import a database already defined as a Sub-License, the TAGs/Plants are automatically overwritten by the ones defined in the new Zipped database
- the database imported is valid only for 15 days
- if you add TAGs to the model of a Sub-License, these new TAGs will be saved/registered in the local database
- the Sub-License database is never saved
- sub-licensing does not work with Demo version

SECTION 3: BELSIM LICENSEMANAGER

3.1 STARTING LICENSEMANAGER

To start the LicenseManager, you can either:

- select LicenseManager Windows Start Menu (**Start Menu | Programs | Vali4 | LicenseManager**)
- double-click the executable "LicenseManager.exe" in the **bin** sub-folder of Belsim installation folder

At start up, the application will require your identification (name and password) and the address of the LicenseServer (IP address or computer name); This address can be selected using the drop-down list. Leaving the address empty will select the current computer as LicenseServer (localhost).



FIGURE 2 - CONNECTION TO LICENSEMANAGER

IP addresses can be added and removed using Edit button. This will open a new window allowing to add and remove IP addresses/computer names, and selecting the default IP address, that will be used by all Belsim applications. ValiScheduler and ValiManager offer the choice of the LicenseServer that will be connected:

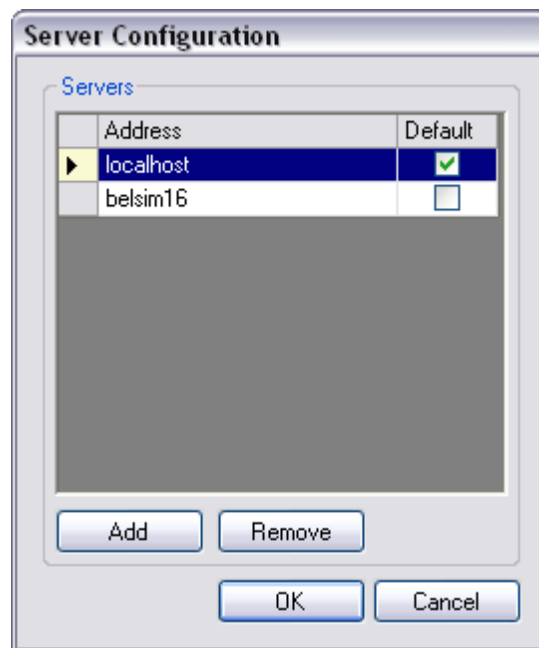


FIGURE 3 - CHOICE OF SERVER

The settings of LicenseManager are accessed via a notebook that has 8 pages:

- General: basic information about available licenses
- ValiManager: list of ValiManager applications
- ValiPerformance list of ValiPerformance applications
- Tag: list of tags that have been defined so far
- Plant: list of plants that have been defined so far
- User: list of users and their associated rights
- Maintenance: back-up, restore and server shut-down
- Handler: list of applications that have required a license

3.1.1 Resetting LicenseServer

In case of bad manipulation, the LicenseServer can be locked to prevent abnormal use of the available licenses. For instance, changing the date time of your computer, restarting the licenseserver without its database files will lock the LicenseServer.

To unlock your LicenseServer, please contact Belsim hotline support to obtain a password then follow the next steps:

- open Windows Explorer
- browse to the installation directory of the LicenseServer ([InstallDir]\LicenseServer\BelsimWS)
- edit the file "LicenseServer.ini",
- add at the end of the file the following lines:

```
[RESTART]
password=*****
```

where ***** is the password provided by Belsim hotline support

- save the file
- open LicenseManager to test the reset was successful

3.2 THE GENERAL PAGE

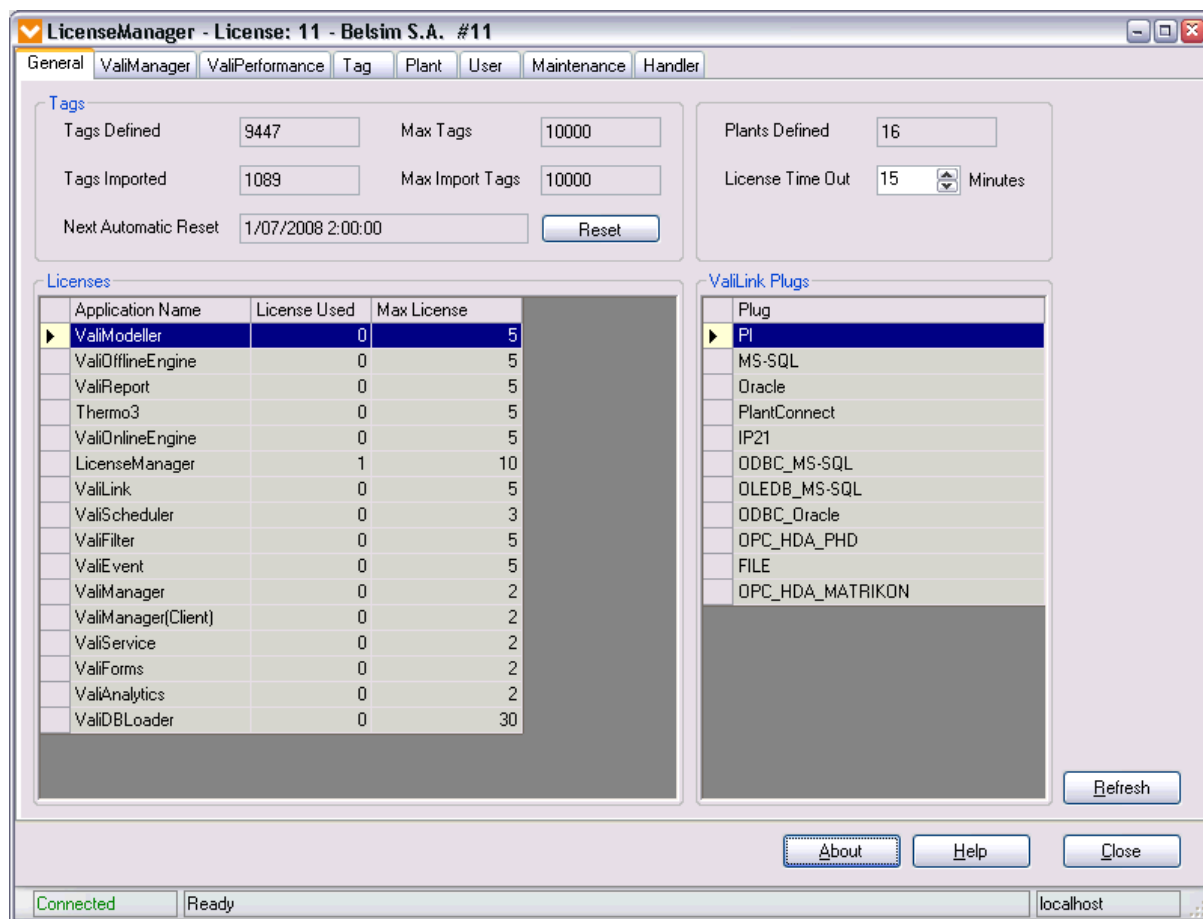


FIGURE 4 - GENERAL PAGE OF LICENSEMANAGER

The **General** page displays the basic information about available licenses:

- Tags defined: number of tags registered in the LicenseServer
- Tags imported: number of tags imported since the last reset
- Next Automatic Reset: date at which the number of imported tags will be reset
- Plants defined: number of plants registered in the LicenseServer
- License Time Out: time-out after which a license can be reallocated when the application is inactive
- Licenses: List of licensed software modules and number of active applications
- ValiLink Plugs: List of licensed ValiLink plugs

Use the **Reset** button to manually reset the number of imported tags. (A password must be requested from Belsim).

Use the **Refresh** button to refresh the information.

3.3 THE VALIMANAGER PAGE

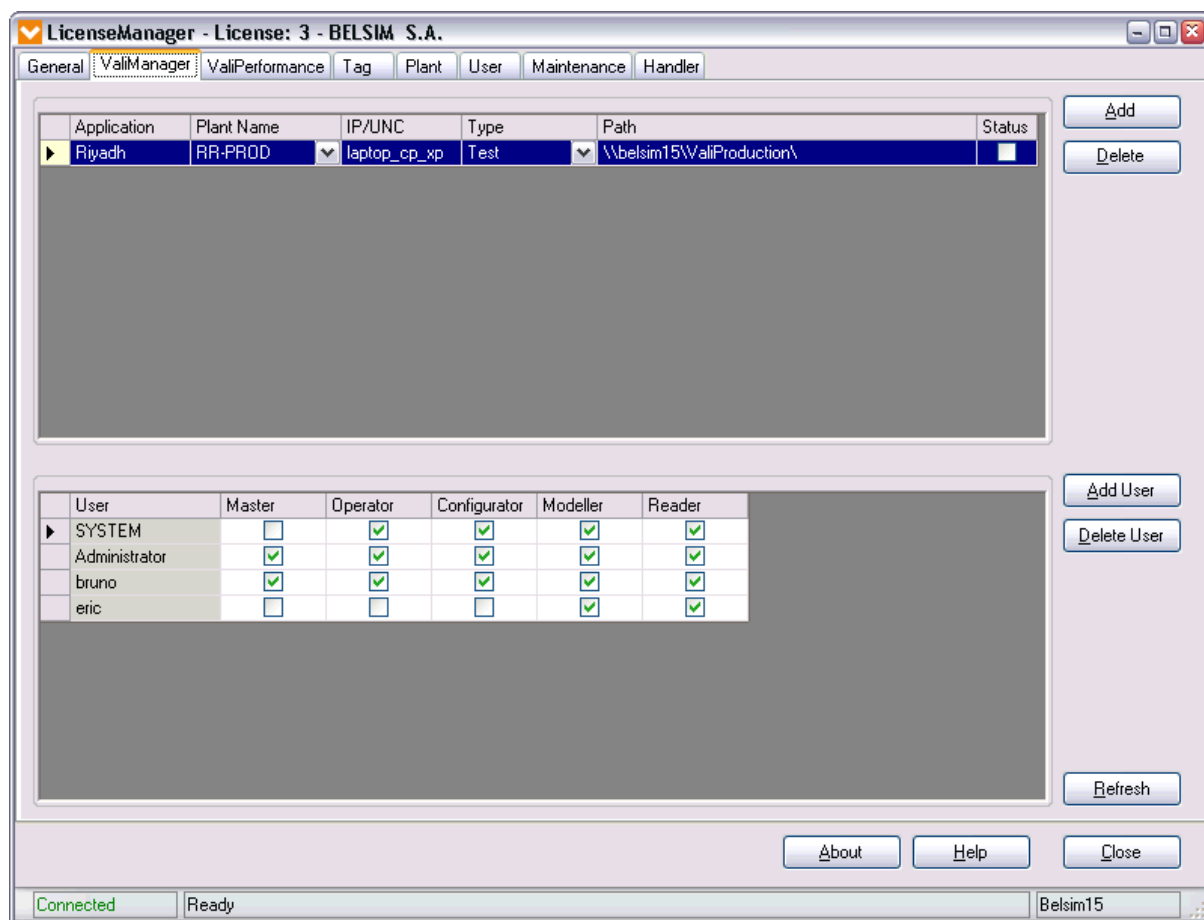


FIGURE 5 - THE VALIMANAGER PAGE OF LICENSEMANAGER

The **ValiManager** page displays the list of ValiProduction applications to be processed by ValiService.

The required settings to define a ValiProduction application are:

- a name
- the plant name giving access to ValiDatabase
- the IP/UNC of the computer with the ValiService that will process the application
- the path where the application files will be saved (path of the application)
- the type of application: [Production|Test]

If the status of the application is set to ON, the application will start to be processed immediately.

After creating a ValiProduction application, the authentication settings must be defined according to a number of roles:

- Master: gives full access to the application including runs approval and changes to official model but not to configuration panel neither to ValiModeller
- Operator: enables to modify settings (ValiMask) and manual corrections on official runs and to ask for reruns
- Configurator: gives access to configuration panels
- Modeller: enables to work in manual mode but not to save as Official model or Official run
- Reader: can only see results (overview of processed runs and access to ValiReport)

The Add Application button is used to define a new application in the LicenseServer. Some information must be provided to create a new application:

- Application name: a unique name identifying the application
- Server name: the name of the computer where ValiService is deployed
- Application Type: (Production or Test). A test application doesn't write back values in the database (ValiLink Out)
- Application Path: the path of the application
- Starting Date: the start date of the application
- Plant Name: the plant name giving access to the ValiDatabase
- Status (ON/OFF): if set to OFF, ValiService will not process this application
- Folders Creation: if checked, all folders and files needed to run an application will be created

FIGURE 6 - THE CREATION OF A VALIPRODUCTION APPLICATION

The Refresh key is used to refresh the complete list of ValiProduction application according to their

last modifications at the License Server level.

3.4 THE VALIPERFORMANCE PAGE

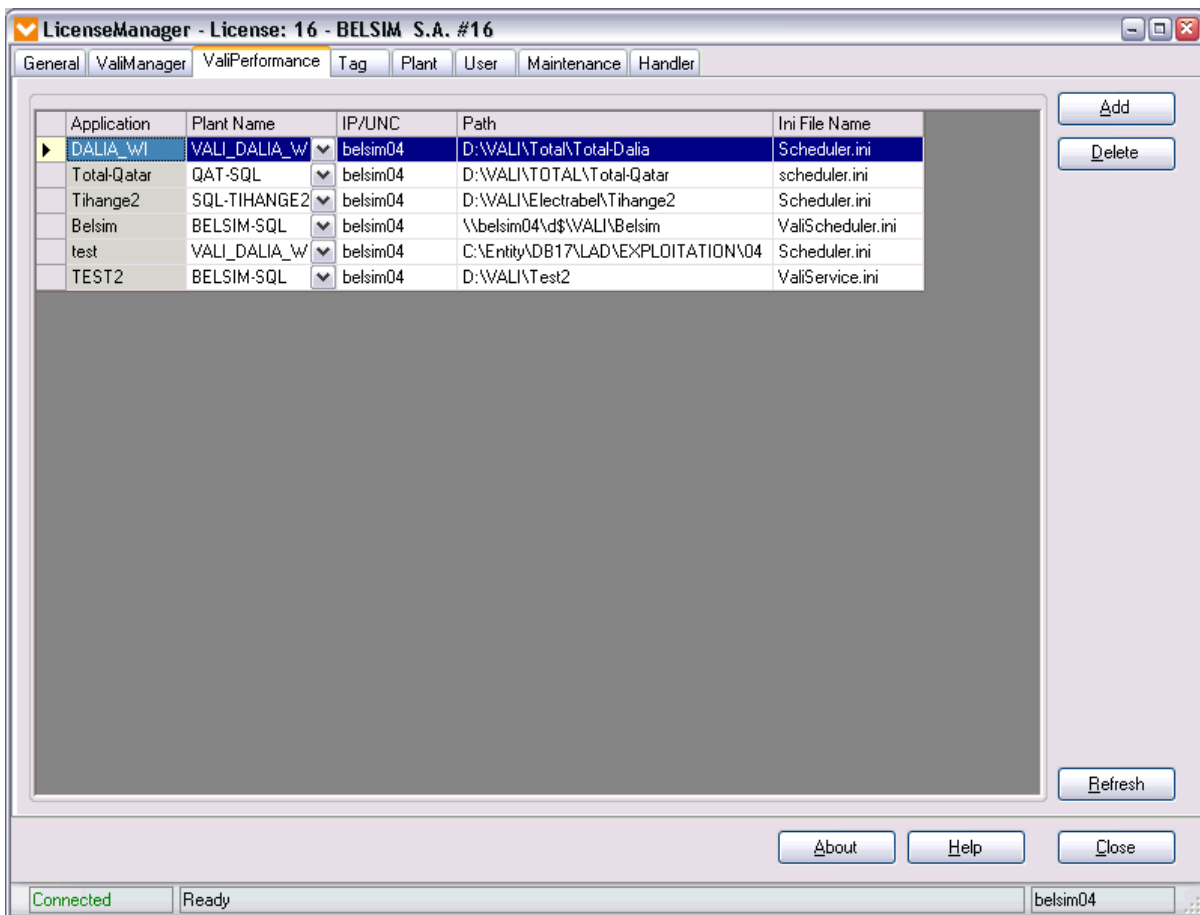


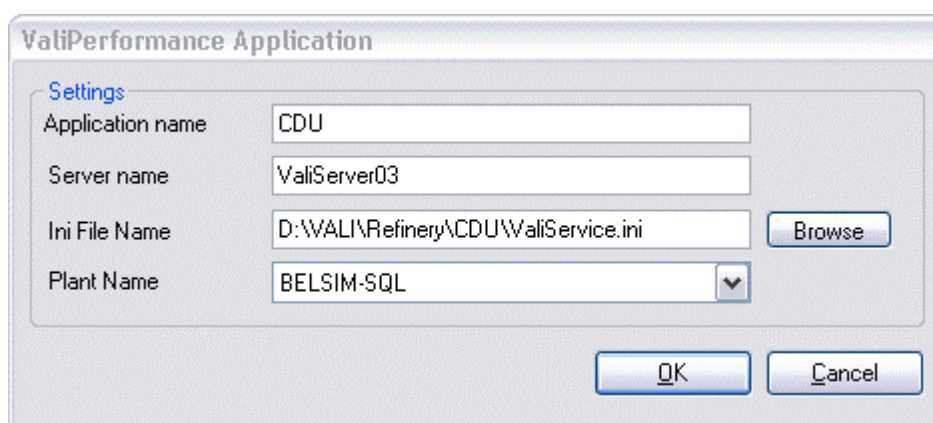
FIGURE 7 - THE VALIPERFORMANCE PAGE OF LICENSEMANAGER

The **ValiPerformance** page displays the list of ValiPerformance applications to be processed by ValiService.

The required settings to define a ValiPerformance application are:

- a name
- the plant name corresponding to the ValiDatabase used by the application
- the IP/UNC of the computer with the ValiService that will process the application
- the path and file name of ValiScheduler configuration file

To add a ValiPerformance, press the Add Application button and enter the required settings:



The screenshot shows a Windows-style dialog box titled "ValiPerformance Application". Inside the dialog, there is a section labeled "Settings" with a blue header. Below this header, there are four labeled text input fields: "Application name" containing "CDU", "Server name" containing "ValiServer03", "Ini File Name" containing "D:\VALI\Refinery\CDU\ValiService.ini", and "Plant Name" containing "BELSIM-SQL". To the right of the "Ini File Name" field is a "Browse" button. At the bottom right of the dialog are "OK" and "Cancel" buttons. The "Plant Name" field has a small downward arrow icon on its right side, indicating it is a dropdown menu.

FIGURE 8 - VALIPERFORMANCE APPLICATION CREATION

The Refresh key is used to refresh the complete list of ValiPerformance application according to their last modifications at the License Server level.

3.5 THE TAG PAGE

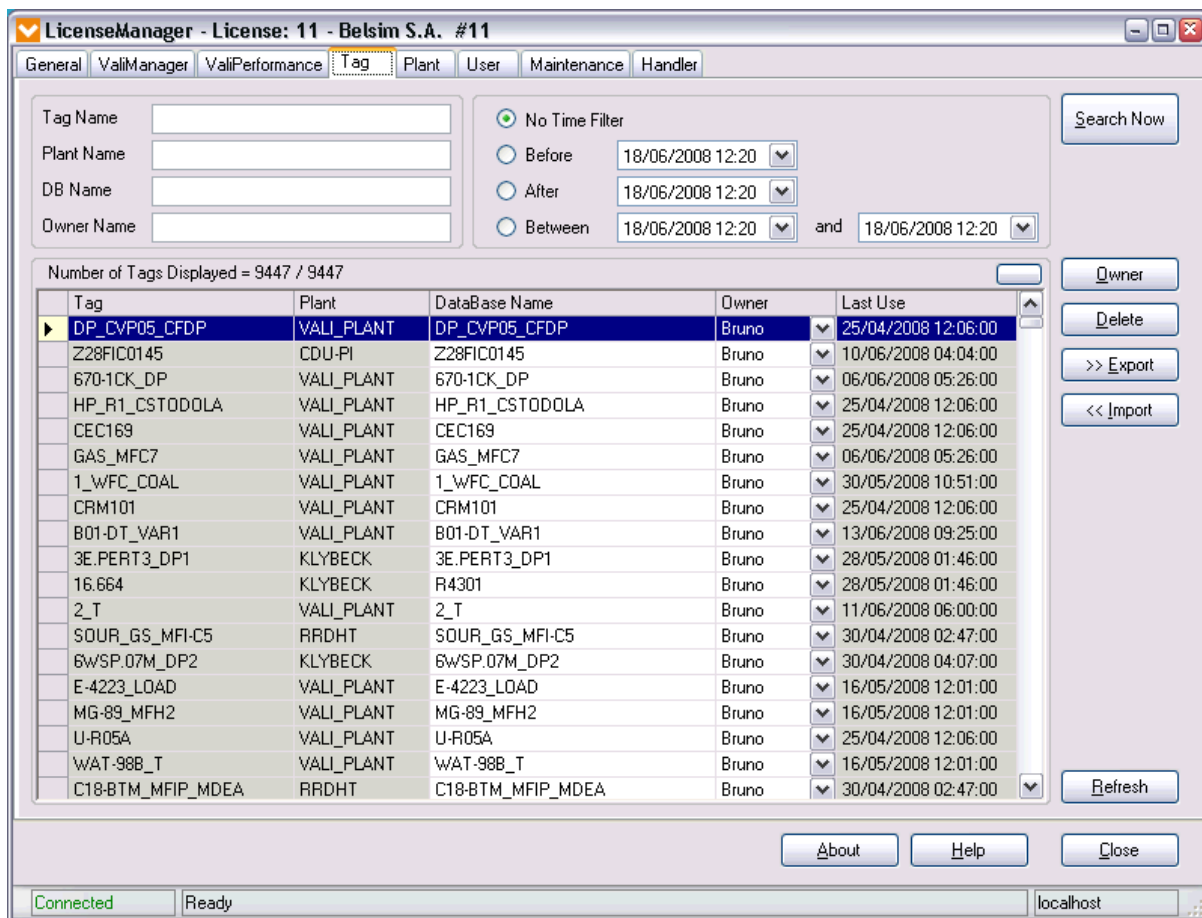


FIGURE 9 - THE TAG PAGE OF LICENSEMANAGER

The **Tag** page displays the list of tags that have been defined together with:

- the plant to which that are associated
- the owner of the tag
- the Dbname (Data Base Name) associated to the TAG
- the date at which the Tag was last referred by an application

The two groups of settings above the Tag list can be used to limit the number of tags displayed in the Tag list by specifying their name, associated plant or owner, and according by their date of last use. You can combine the various conditions and use wildcards for the names (tag, plant, owner).

After any change in these settings use the **Search Now** button or press **Enter** key to reprocess the list according the new conditions.

The **Owner** button enables to reallocate the selected tags to another user.

The **Delete** button is used to delete the selected tags.

The **Export** button is used to export the list of selected tags as an ASCII file with the data included in each column:

- the tag name
- the plant name
- the Dbname (if any)
- the owner
- the last use date time

The **Import** key is used to import a similar list. Only the first 3 columns will be used for the import, they must include the following information:

- the tag name
- the plant name
- the Dbname (if any)

Please note that in a given month you cannot import more tags than the number of tags you have been licensed.

The **Refresh** key is used to refresh the complete list of tags according to their last modifications at the License Server level.

Attention: You will use the Export/Import facility for example to update the tag name / Dbname association via a Text Editor.

3.6 THE PLANT PAGE

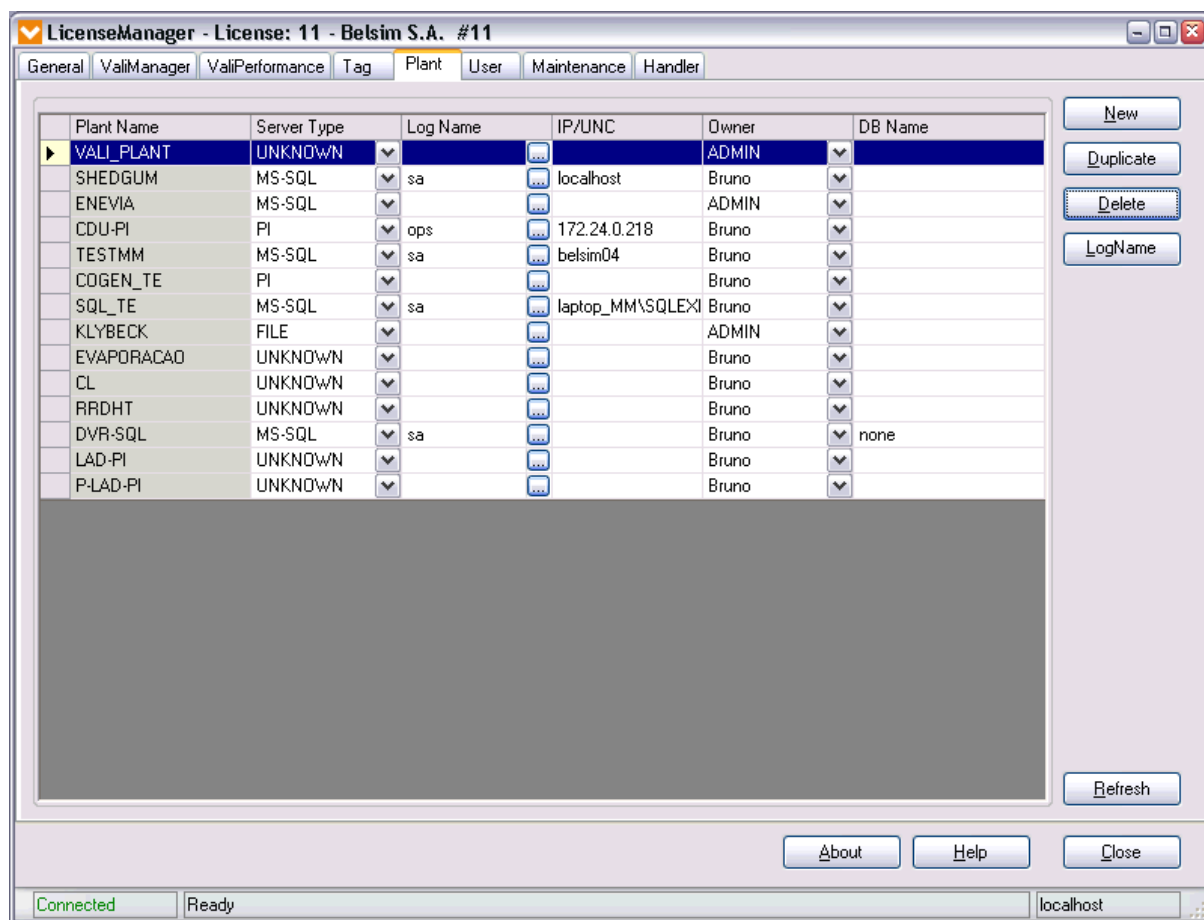


FIGURE 10 - THE PLANT PAGE OF LICENSEMANAGER

The **Plant** page displays the list of plants together with:

- their owner
- the type of server (see [ValiLink](#) documentation)
- the Log name to be used when connecting to the corresponding server
- the IP/UNC address of the corresponding server
- the DB name

The **New** and **Duplicate** button allows creating a new plant either from scratch or by using the data of an existing plant.

The **LogName/Password** button is used to modify the log name password associated to a given plant.

The **Delete** button enables to delete a plant. You can only delete a plant if no tags are associated to it.

The **Refresh** key is used to refresh the list of plants according to their last modifications at the License

Server level.

3.7 THE USER PAGE

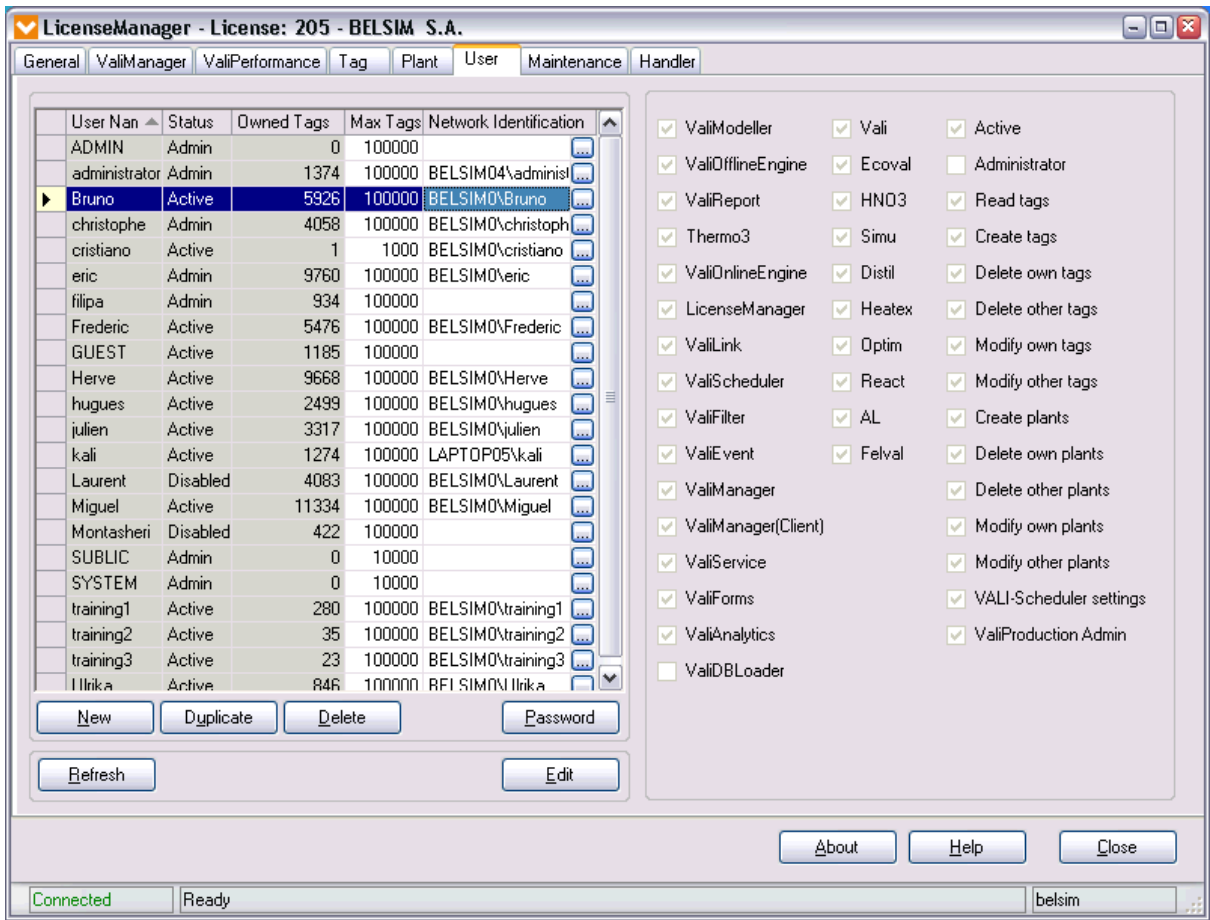


FIGURE 11 - THE USER PAGE OF LICENSEMANAGER

The **User** page displays the list of users together with:

- their status:
 - Admin: the user has administration rights
 - Active: the user is defined and enabled
 - Inactive: the user is defined but currently disabled; he will not be authorized to access any of the licensed software modules
- the number of tags they currently own
- the maximum number of tags they can define
- the network identification (interaction with Windows Active Directory)
- their rights (displayed on the right panel)

The **Password** button is used to modify the password associated to a given user. Administrator status is required in order to change the password of other users.

The **New** and **Duplicate** button allows creating a new user either from scratch or by using the data of an existing user (administrators only).

The **Delete** button enables to delete a user (administrators only). You can only delete a user if he does not own any tag.

When creating a new user, a Network Identifier will be automatically detected using Windows Active Directory. The Network Identification can be changed manually by clicking on the button  in the "Network Identification" column.

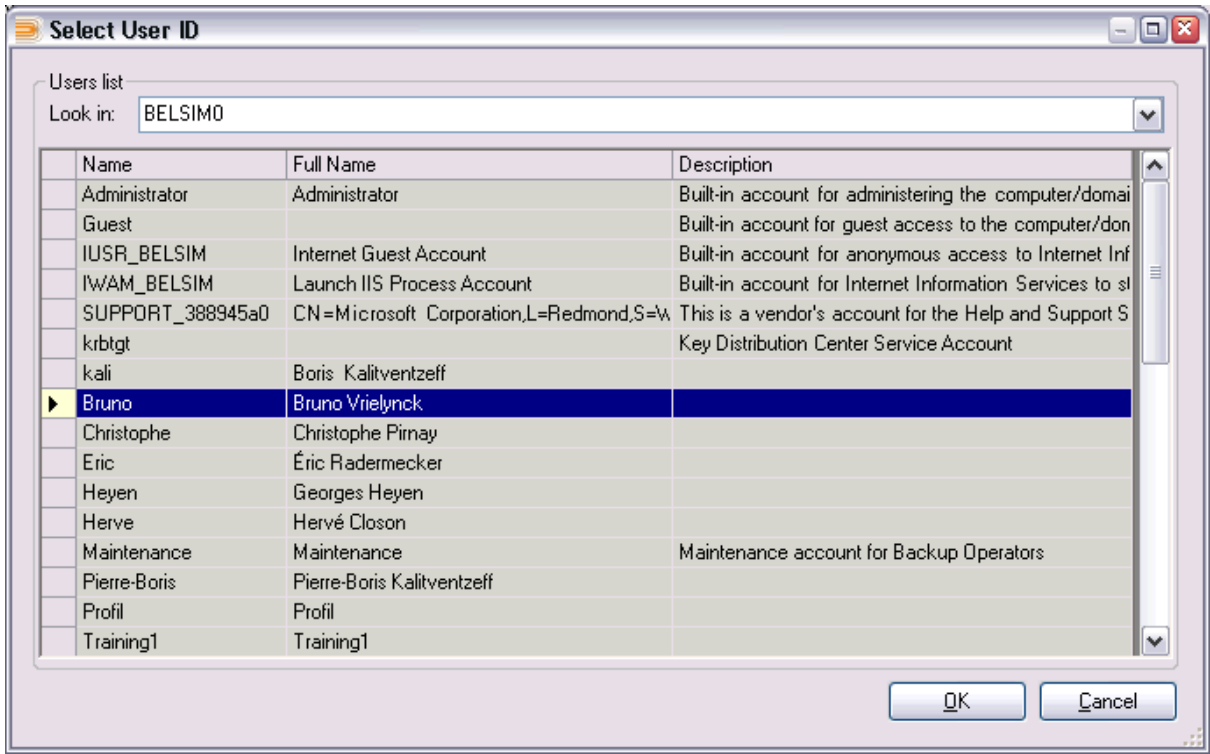


FIGURE 12 - NETWORK IDENTIFICATION SELECTION

The **Edit** button is used to modify the rights of a user.

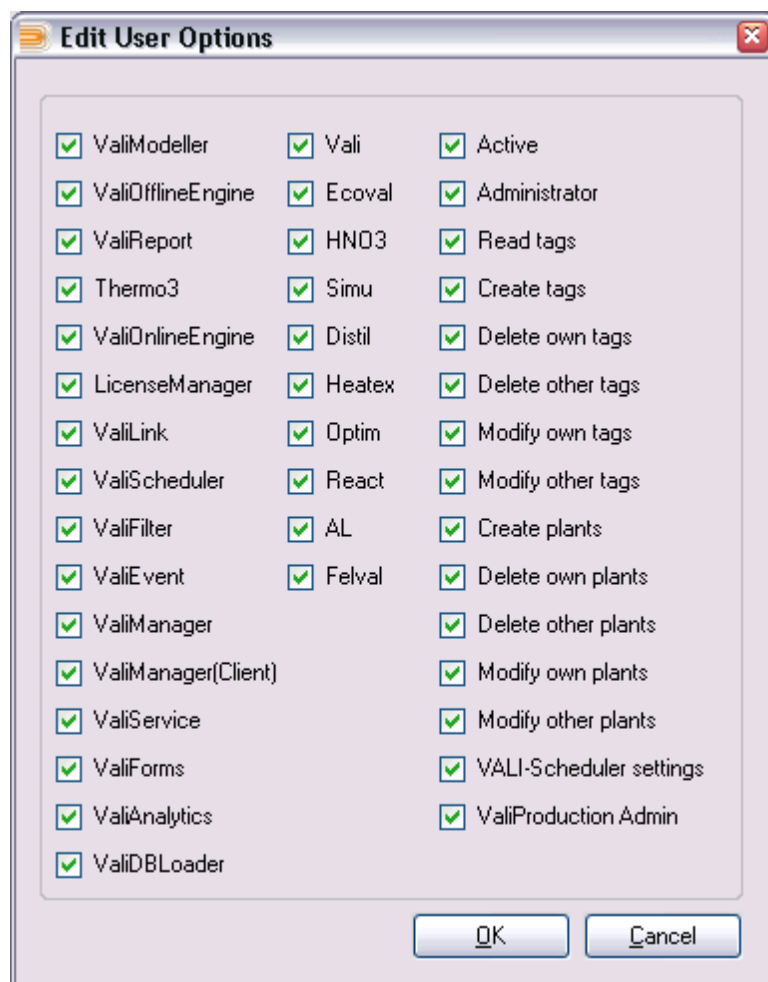


FIGURE 13 - EDITING USER RIGHTS

3.8 THE MAINTENANCE PAGE

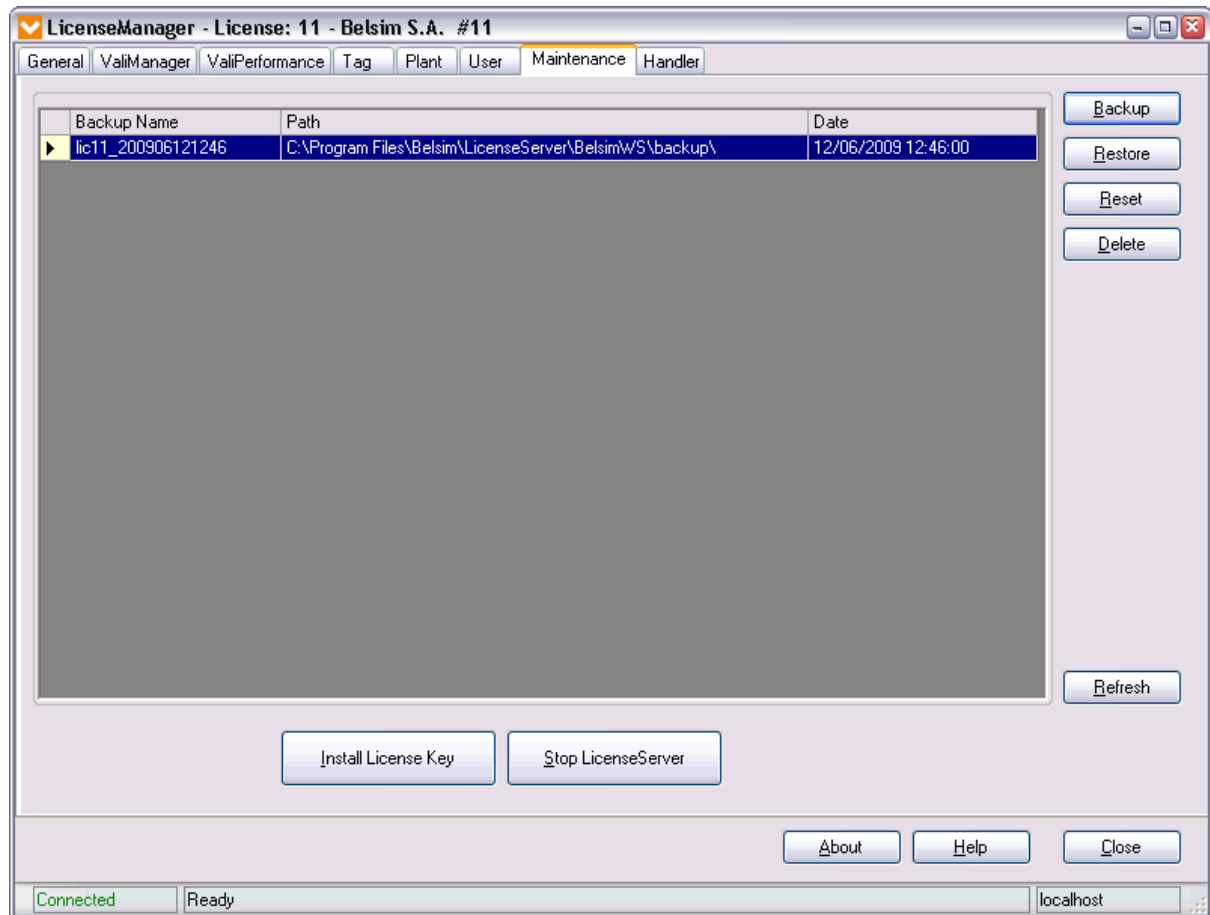


FIGURE 14 - THE MAINTENANCE PAGE OF LICENSEMANAGER

The **Maintenance** page displays the list of back-up of the database together with:

- the path of the back-up file
- the data of the back-up

Use the **Back-up** button to create a new back-up of the database. You can create as many back-up as you want.

Use the **Restore** button to restore the selected bac-up. Please note that you can restore only one back-up per month. A special password must be requested from Belsim to overcome this limitation through the **Reset** button

Use the delete key to delete the obsolete back-ups.

Use the **Stop LicenseServer** button to stop the LicenseServer.

Use the **Install License Key** button to upload a new license key provided by Belsim.

3.9 THE HANDLER PAGE

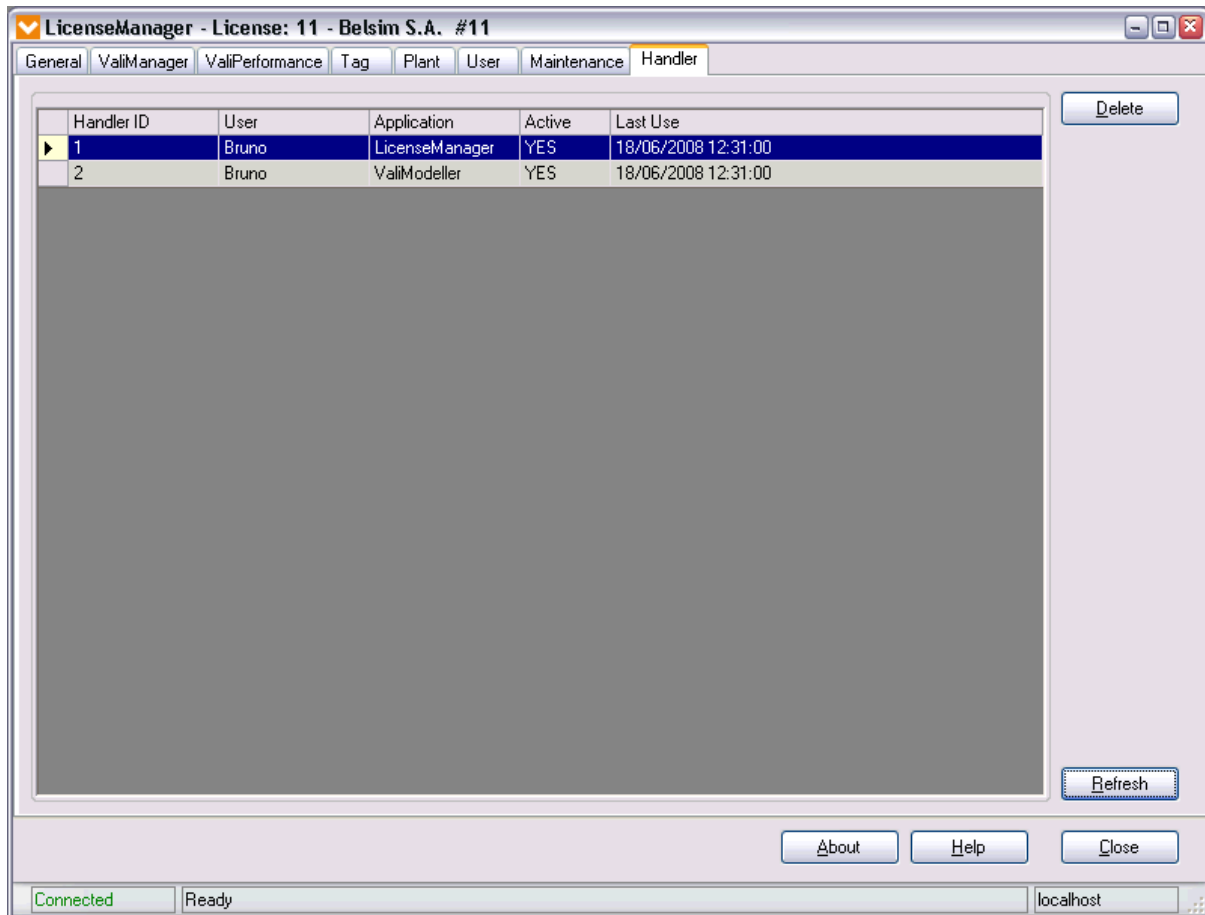


FIGURE 15 - THE HANDLER PAGE OF LICENSEMANAGER

The **Handler** page displays the list of handlers together with:

- the corresponding user
- the application name
- its activity (NO means that the application has not been active for a duration of time greater than the time-out)
- the date and time of the last request to the LicenseServer

The **Delete** button enables to delete a handler in order to free up the corresponding license (administrators only).



belsim

CHAPTER 12

VALIPERFORMANC E

SECTION 1: VALI Software Suite

VALI is a process oriented software aiming at determining and improving production efficiency, productivity and energy consumption while reducing production and maintenance costs.

With VALI, you move from data to information, from information to knowledge and finally from knowledge to action.

FROM DATA TO INFORMATION

Measurements are the essential source of information of VALI (in addition to the process topology and the process management information) and cover all information coming either from the plant (temperatures, pressures, rates, on-line analyses, etc.) or from the laboratory (GC, densities, etc.).

VALI converts the raw measurement data into reliable information by applying data validation technique involving among others heat and mass balances.

FROM INFORMATION TO KNOWLEDGE

From the coherent information, VALI determines accurate and reliable performance indicators for each equipment of the plant (heat rates, rotating equipment efficiencies, reaction's conversions, etc) as well as more global KPIs (Key Performance Indicators) like plant efficiency, plant specific product yields, etc.

VALI pinpoints instrumentation deviation and equipment degradation.

FROM KNOWLEDGE TO ACTION

By generating customised reports for each user (maintenance, control, management, production...), adequate action will rapidly be undertaken. Using validated data for generating warnings strongly reduces false alarms. A clear distinction is made between instrumentation problem and equipment degradation.

1.1 ValiPerformance

ValiPerformance is oriented towards online process performance monitoring, i.e. online reporting of the key performance information of a process such as process yield, reaction selectivity, delta_T approach to the equilibrium, energy consumption, equipment efficiency (e.g. heat transfer coefficient, turbine and compressor efficiency...).

The application calculates detailed heat and mass balances of the units, including utilities (steam and fuel-gas). These balances are calculated using all available measurement redundancy to determine a single set of values consistent with conservation of heat and mass. This data represents the most likely point of actual plant operation.

This information can then be used to, among others:

- Detect performance degradation and take corrective action
- Identify heat loads, heat transfer coefficients and fouling factors on heat exchangers
- Identify heat loads and efficiencies of heaters and furnaces
- Identify power and efficiency of rotating equipment
- Identify power required for pumps
- Identify conversion (and catalyst deactivation when suitable) in reactors
- Identify power and efficiency of rotary machines
- Calculate various Key Performance Indicators such as (this list is not exhaustive):
 - Yields
 - Efficiencies
 - H₂ partial pressure or Steam/Carbon ratios
 - Sulphur conversion or recovery
 - Global energy efficiency
 - Specific energy consumption

Compound mass balances, heat balance and pressure drop balances are generated around each equipment. Liquid/Vapour separators also include L/V equilibrium constraints. Reactors are modelled either by atomic balances or according to defined reaction schemes. Chemical equilibrium constraints are also taken into account where necessary.

1.2 VALI modules

- ValiModeller : a multi-purpose modelling tool intended to develop and maintain one or several process models. ValiModeller can be used for offline validation and engineering studies
- ValiAnalytics, a Graphical User Interface used for sensitivity analysis of your model and your results
- ValiReport : a tool integrated in MS Excel® and designed for the visualisation and reporting of results and that is configurable according to user needs
- ValiOnlineEngine: : designed to be implemented online to generate the results of validation on the basis of online data sets and for models developed with ValiModeller
- ValiFilter : a data filtering module, which ensures an optimal operation of VALI through a pre-processing of raw measurement data
- ValiEvent : an add-on that analyses the online measurement data in a fixed time interval and reports on the detected events
- ValiLink : data connection module between VALI and an external system (can be PIMS, DCS, LIMS...)
- ValiManager : an application that runs on top of VALI and that handles different aspects of production accounting as access authorizations, periodic runs, database storage, model versioning, etc. ValiManager deals with total site production accounting on a stated period of time (typically one day)



belsim

CHAPTER 13

VALILINK

SECTION 1: INTRODUCTION

ValiLink is a generic application allowing to transfer information between VALI and external systems such as PIMS, DCS or more generally data bases. The currently supported systems are:

- PI (OSIsoft ©)
- PlantConnect (ABB ©)
- InfoPlus21 (AspenTech ©)
- WinIS (Siemens ©)
- Microsoft SQL Server 7.0 or later (using SQL Server .NET Data Provider)
- Oracle Database 8.0 or later (using Oracle .NET Data Provider)
- PHD (Honeywell ©)
- any database compliant with the OLE DB .NET Data Provider.
- any database compliant with the ODBC .NET Data Provider.
- any database compliant with the OPC HDA standard
- ASCII files.

ValiLink can be used both for reading or writing data from and to the external system. Typically you will use ValiLink before starting a ValiEngine (online or offline) to get data from one or several data sources and you will start it again after the run to store the results.

Attention: WinIS plug only supports reading from WinIS system

ValiLink can be called directly from ValiReport and ValiScheduler. It can also be launched manually or via another application.

Three different operation modes are available:

- ValiLink directly reads the list of tags to be acquired (or written) in the bls file. This procedure can only be used with direct access system, not with SQL like databases
- ValiLink reads the description of operations to be performed in an INI file, which will in turn refer to ASCII files with the list of tags to be read from (or written to) the external system
- ValiLink reads the description of operations to be performed in an INI file, and reads in the bls file the list of tags to be read from (or written to) the external system

The INI file can easily be configured using ValiLinkBuilder application or can be edited using a text editor.

1.1 USING VALILINK TO GET DATA

CONFIGURATION VIA THE MODEL FILE

- ValiLink receives as arguments the name of the bls file as well as the operation (IN for input) and the time window
- the name of the server (external system) to be connected to is read from the bls file (defined through the plant associated to each tag), while the userid and password to be used are read from LicenseServer
- the TAG values are then read from the server and written in a file called Valilink.me

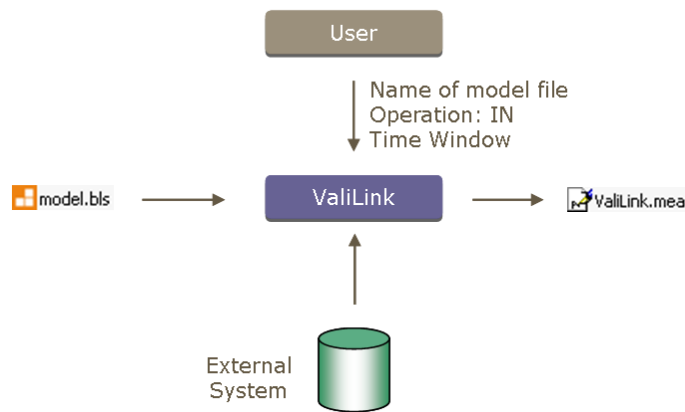


FIGURE 1 - CONFIGURING VALILINK INPUT VIA A MODEL FILE

CONFIGURATION VIA AN INI FILE AND A MODEL FILE

- ValiLink receives as arguments the name of the INI file as well as the time window
- the name of the server (external system) to be connected to is read from the INI file, while the userid and password to be used are read from LicenseServer
- the INI file specifies for each action the name of a model file where the list of tags to be acquired will be read
- the TAG values are then read from the server and written in the Output File specified in the INI file (e.g. ValiLink.mea)

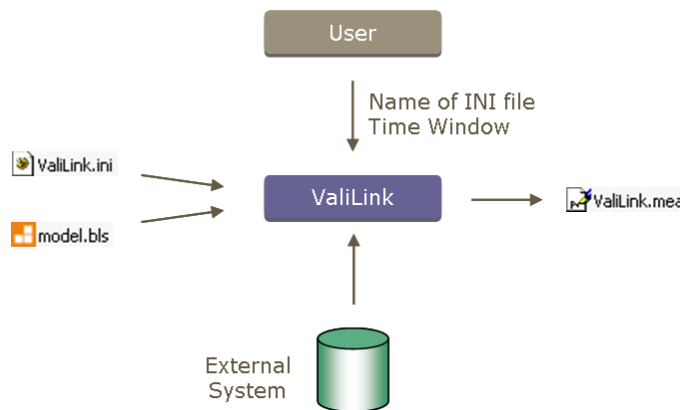


FIGURE 2 - CONFIGURING VALILINK INPUT VIA AN INI FILE AND A MODEL FILE

CONFIGURATION VIA AN INI FILE

- ValiLink receives as arguments the name of the INI file as well as the time window
- the name of the server (external system) to be connected to is read from the INI file, while the userid and password to be used are read from LicenseServer
- the list of tags to be acquired is read from an ASCII file specified in the INI file

- the TAG values are then read from the server and written in the output file specified in the INI file

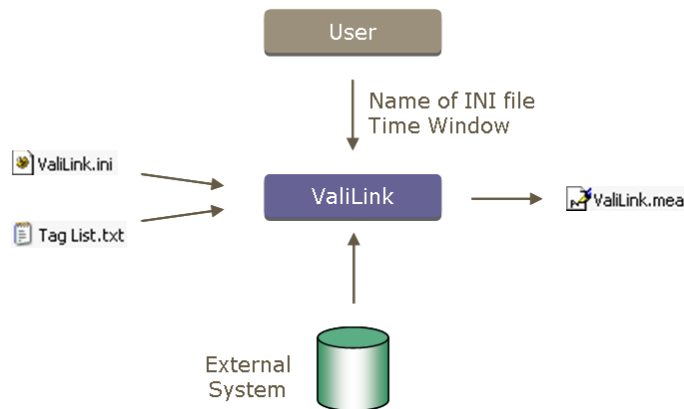


FIGURE 3 - CONFIGURING VALILINK INPUT VIA AN INI FILE

1.2 USING VALILINK TO WRITE DATA

CONFIGURATION VIA THE MODEL FILE

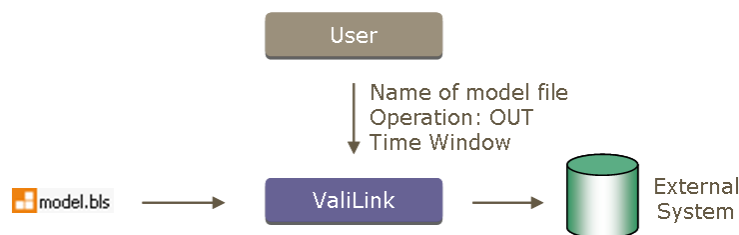


FIGURE 4 - CONFIGURING VALILINK OUTPUT VIA A MODEL FILE

- ValiLink receives as arguments the name of the bls file as well as the operation (OUT) and the time window
- the name of the server (external system) to be connected to is read from the bls file (defined as the plant associated to each tag), while the userid and password to be used are read from LicenseServer
- the TAG values are then read from the bls file and written to the external system

CONFIGURATION VIA AN INI FILE

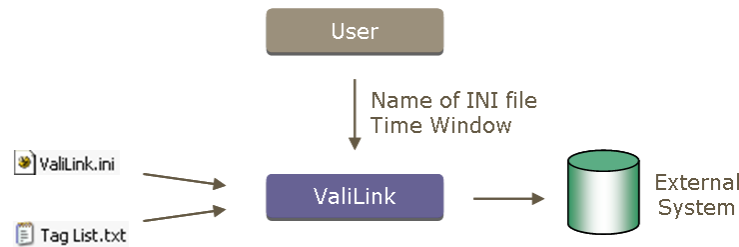


FIGURE 5 - CONFIGURING VALILINK OUTPUT VIA AN INI FILE

- ValiLink receives as arguments the name of the INI file as well as the time window
- the name of the server (external system) to be connected to is read from the INI file, while the userid and password to be used are read from LicenseServer
- the list of tags and their values are read from the InputFile specified in the INI file
- the TAG values are then written to the server

1.3 TAGNAMES VERSUS DBNAMES

In some cases the names of the tags in your Vali model will be different from the names of the tags as defined in the external system.

This is solved by associating to each Vali tag two names:

- the tagname which is the usual tag name defined in the VALI model
- the Dbname which is the name used in the external system. This second name will be stored in the LicenseManager and will therefore be available for all models referring to the same tag

The Dbname is defined when creating the tag with ValiModeller. It can be redefined afterwards using LicenseManager or ValiModeller.

When getting (writing) values from (to) an external system, ValiLink will use the Dbname (if defined) in place of the Tagname when communicating with the external system. Moreover, ValiLink will add any defined prefix or suffix to the Dbname (or tagname if no Dbname is defined) when writing the values to the external system.

SECTION 2: RUNNING VALILINK

2.1 STARTING VALILINK

ValiLink has been designed to be automated through ValiReport or ValiScheduler. However it can also be automated through other applications.

To start ValiLink, start ValiLink.exe in <installdir>\bin folder (by default Program Files\Belsim\bin) with the following arguments.

Specification via the bls file:

- the keyword **-BLS**
- the name of the model (bls) file
- the keyword **IN** or **OUT**
- one date with a fixed format
- one duration value, which can be negative
- the physical units of the duration: S, s, SEC, Sec, sec, M, m, MIN, Min, min, Hour, H, h, HOUR, Hour, hour, Day, D, d, DAY, Day, day

Specification via an ini file:

- the name of an information file (ini file) describing the actions to be performed
- one date with a fixed format
- one duration value, which can be negative
- the physical units of the duration: S, s, SEC, Sec, sec, M, m, MIN, Min, min, Hour, H, h, HOUR, Hour, hour, Day, D, d, DAY, Day, day

Example:

```
C:\Program Files\Belsim\bin\ValiLink.exe -BLS C:\Demo\coal.bls IN 20040826_162839 24 H
C:\Program Files\Belsim\bin\ValiLink.exe C:\Demo\SomeActions.ini 20040826_162839 24 H
```

2.2 SPECIFYING THE VALIDATION TIME WINDOW

The date and time format used in the argument of the command line is YYYYMMDD_hhmmss

Example:

20040826_162839 corresponds to 26 August 2002 at 16:28:39 (4:28:39 pm)

The Time Window of the validation run is delimited by two dates: BeginDate and EndDate.

If the duration value is **positive**, then the given date is the BeginDate and the EndDate will be calculated as:

$\text{EndDate} = \text{BeginDate} + \text{Duration}$

If the duration value is **negative**, then the given date is the EndDate and the BeginDate will be calculated

as:

$\text{BeginDate} = \text{EndDate} + \text{Duration}$

Examples:

```
C:\Belsim\BinDemo\ValiLink.exe SomeActions.ini 20080327_120000 24 H
```

The BeginDate will be March 27, 2008 at 12:00:00

The EndDate will be March 28, 2008 at 12:00:00

```
C:\Belsim\BinDemo\ValiLink.exe SomeActions.ini 20080327_120000 -1 H
```

The BeginDate will be March 27, 2008 at 11:00:00

The EndDate will be March 27, 2008 at 12:00:00

Attention: The date will automatically be written in the generated output file as RUNDATE field.

2.3 VALILINK STATUS

At the end of its execution, ValiLink produces a file named **ValiLink_log.ini** containing the keyword EXITCODE followed by a numeric code:

- -1: a fatal error did occur
- 0: ValiLink was successful

Example:

```
[EXITCODE]  
ExitCode=0
```

SECTION 3: CONFIGURING VALILINK - VALILINKBUILDER

3.1 STARTING VALILINKBUILDER

To create a new ValiLink file using ValiLinkBuilder, you can either

- Go to **Start Menu | Programs | Belsim | Vali4 | ValiLinkBuilder**
- Go to the bin sub-folder of the Belsim installation folder and double-click on **ValiLinkBuilder.exe**

First, save the ValiLink file (commonly named ValiLink_IN.INI or ValiLink_OUT.INI for measurements reading and writing respectively), then choose the Database type from/to which measurements are read/written:

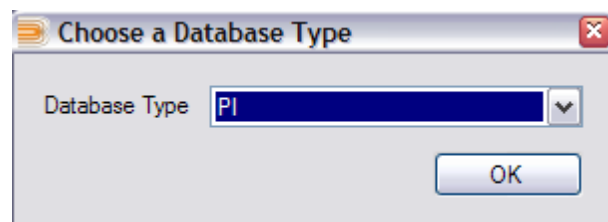


FIGURE 1 – CHOOSE DATABASE TYPE

The ValiLinkBuilder main interface displays the data base type that has been previously chosen:

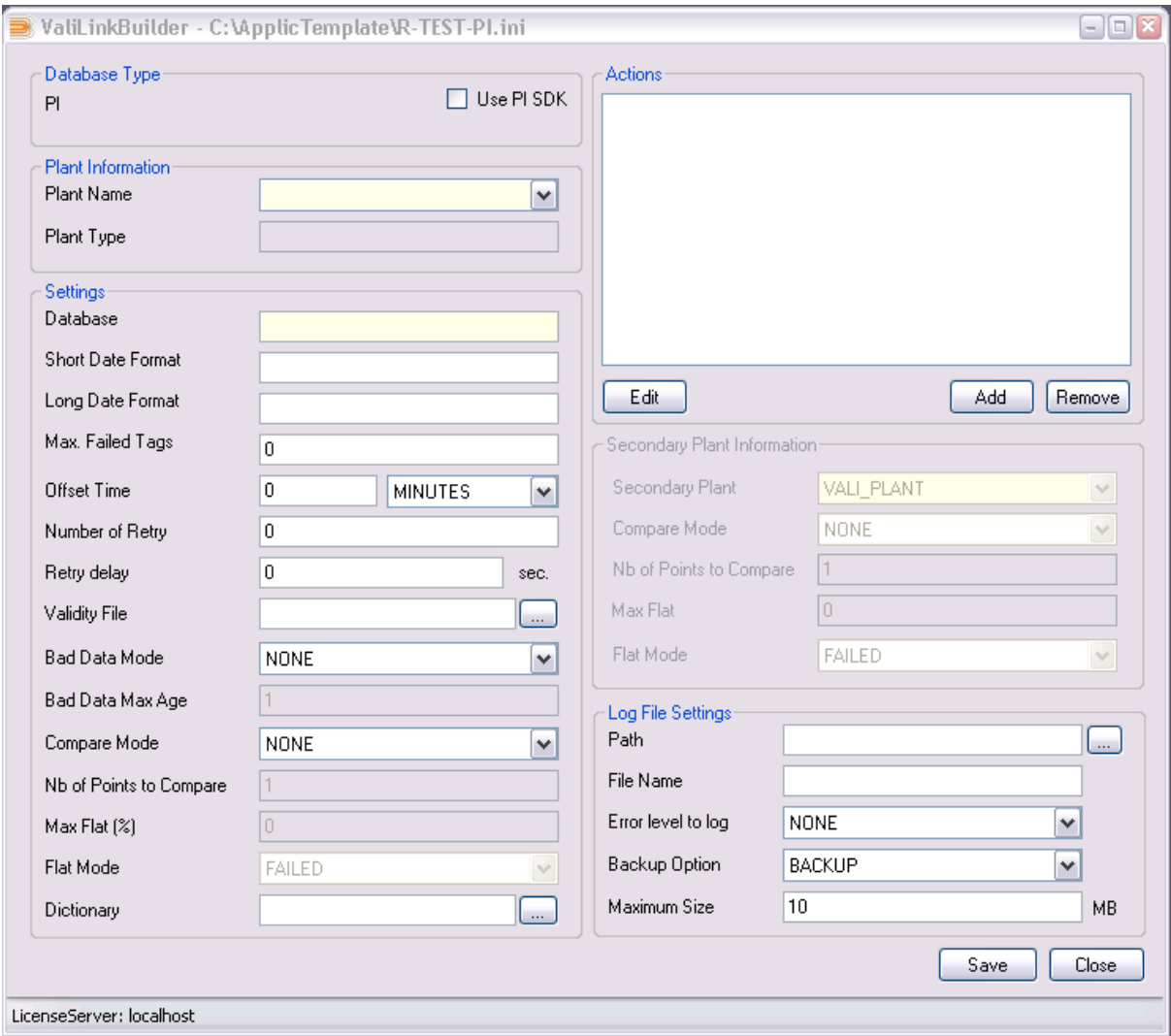


FIGURE 2 – VALILINKBUILDER MAIN WINDOW

To edit an existing ValiLink.INI using the ValiLinkBuilder:

- for measurements reading, go to the ValiScheduler - Tab 'Acquisition' and press the Edit button related to the existing ValiLink file
- for measurements writing, go to the ValiScheduler - Tab 'Storage' and press the Edit button related to the existing ValiLink file

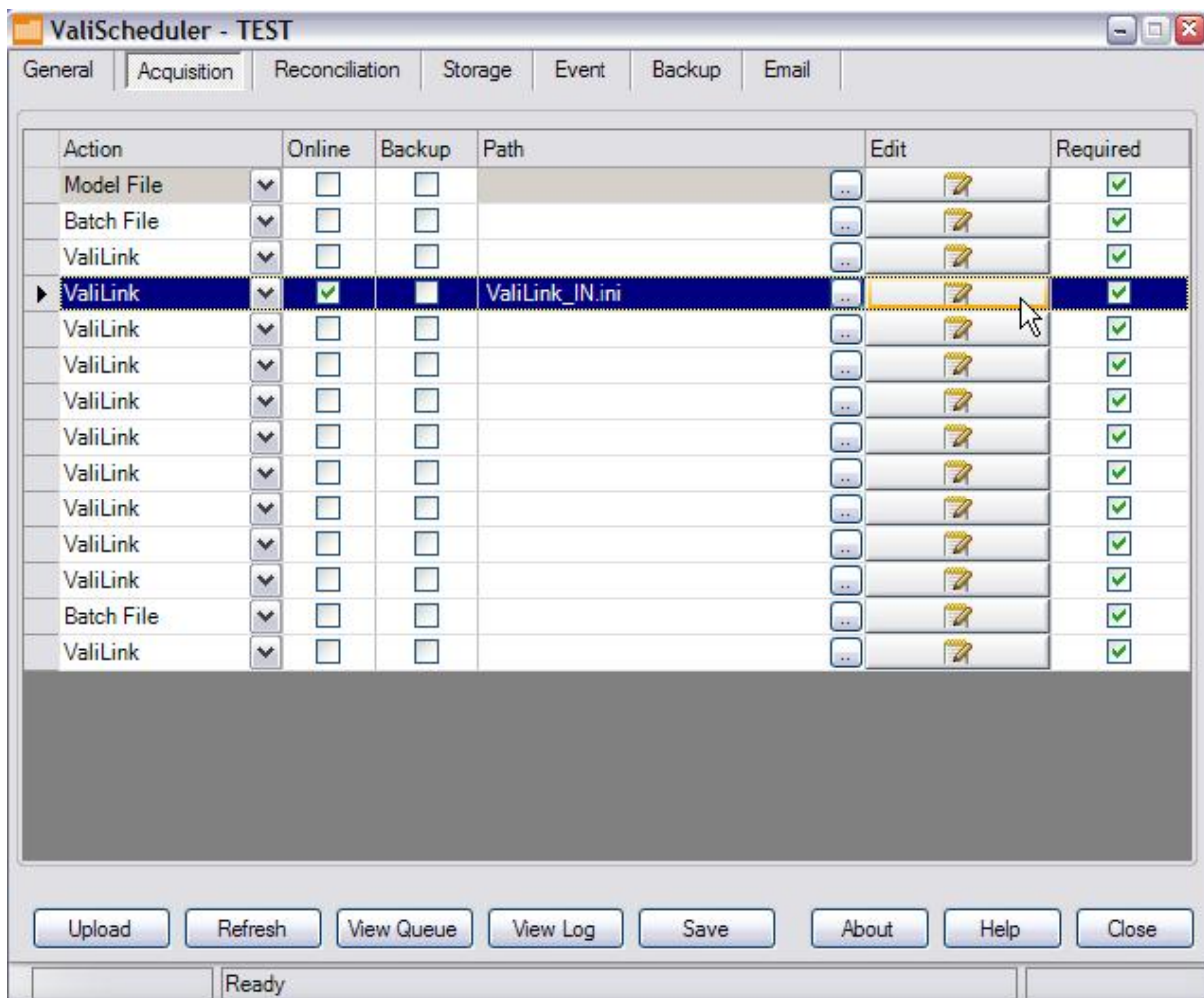


FIGURE 3 – VALISCHEDULER ACQUISITION TAB

3.2 GENERAL SETTINGS

The **main window** allows specifying the general settings for ValiLink.

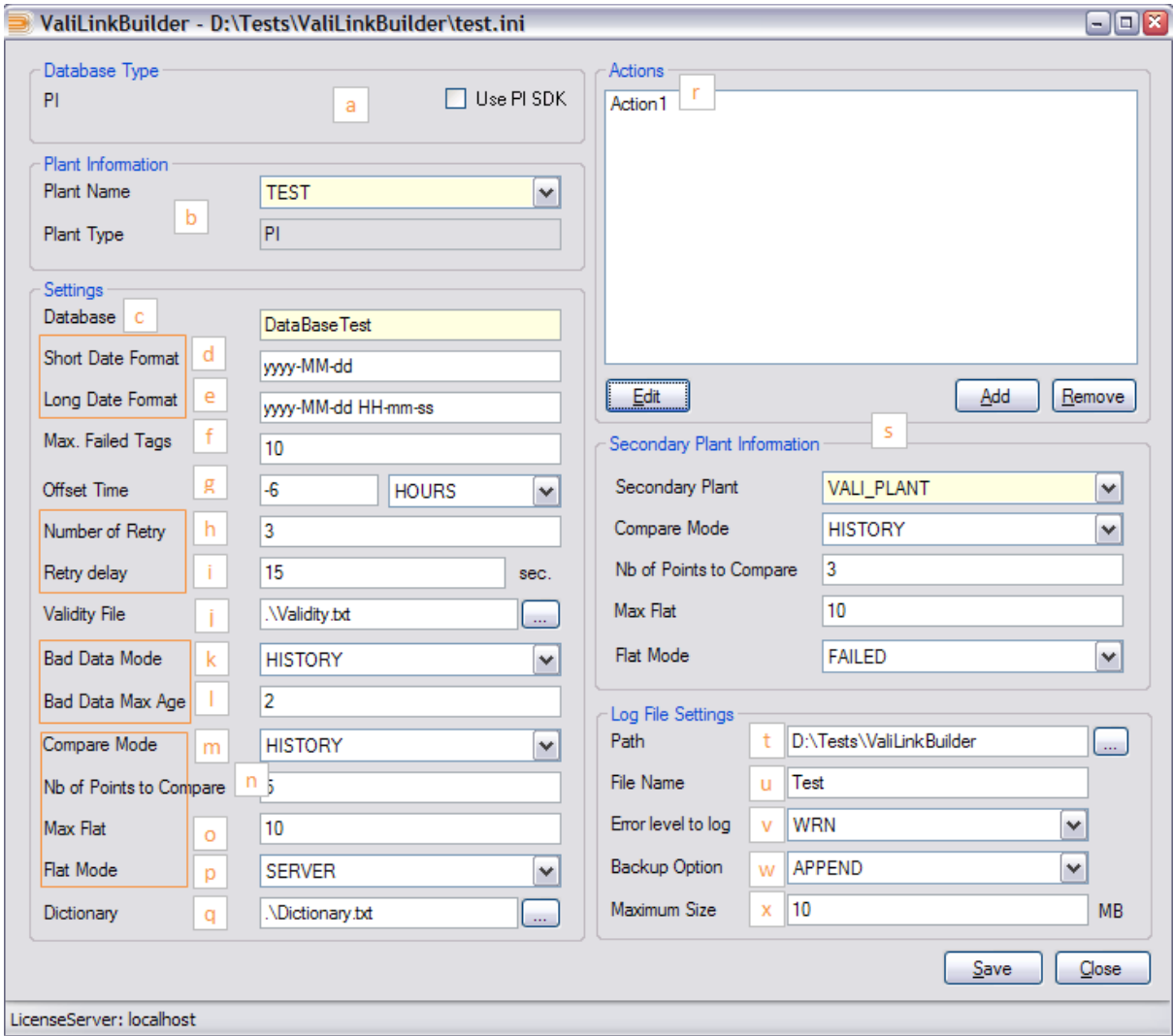


FIGURE 4 – VALILINKBUILDER MAIN WINDOW DESCRIPTION

The **Database Type** [a] contains the type of database chosen when creating a new ValiLink file. With PI (OSisoft ©), there are two ways of querying the database:

- through PI Software Development Kit (SDK),
- or through PI Advanced Programming Interface (API).

PI API is the default interface. To use PI SDK for connection, the "use PI SDK" checkbox has to be checked.

The **Plant information** [b] contains information related to the database server. The Plant Name is to be selected amongst the plants defined in the LicenseManager and provides information related to the external database:

- the type of external database (PI, IP21, PlantConnect, MS-SQL, OLEDB_MS-SQL, ODBC_MS-SQL, Oracle, ODBC_Oracle, OPC_HDA_PHD, OPC_HDA_MATRIKON, FILE)
- the Server (TCP-IP/UNC) of the external database or its Windows Network Name
- the user name to log in the external database server
- the password to log in the external database server

Attention: the database server related to the chosen plant should be of the same type as the one defined at the ValiLink file creation [a].

The **Settings** contains the following information:

- Database **[c]** is the database used by default when no database has been defined in the Actions section **[r]**
- Short Date Format **[d]** is the format of the date used for SQL transactions
- Long Date Format **[e]** is the format of the date and time used for SQL transactions
- Max. Failed Tags **[f]** is the maximum allowed number of tags for which ValiLink is unable to retrieve values. If the maximum allowed number is exceeded, the acquisition step status is failed
- Offset Time **[g]** aims at taking into account the possible time difference in between the ValiScheduler time zone and the external system time zone. The offset time is the time added to the time range (begin and end dates) given as an argument to the ValiLink
- Number of Retry **[h]** is the number of trials processed by ValiLink in case of connection failure with the external system for example
- Retry Delay **[i]** is the time duration in between two trials **[h]**
- Validity File **[j]** gives the path and name of the Validity file which contains the date and time from which given measurement data are to be acquired (Syntax of Validity file: TAGNAME yyyymmdd_hhmmss). Before the mentioned date and time, the specified measurement data are not valid. Note that the path is relative to the ValiLink configuration file
- Bad Data Mode **[k]** is the mode selected for handling bad data (i.e. unavailable data, -999.999, etc.):
 - NONE: The key word BAD is used as the tag value (Example of outputs: TAGNAME BAD)
 - HISTORY: If the data is bad at time T, the values in the database at time T-1 to T-n are checked (n is expressed as a multiplication factor of the aggregation period set for calculation and is defined at section Bad Data Max Age **[l]**). The first valid value that is found, started from T-1, is retrieved and takes the place of the bad data (Example of outputs: TAGNAME 10°C BAD)
 - MEA: If the data is bad at time T, the values in the backup directory at time T-1 to T-n is checked (n is expressed as a multiplication factor of the aggregation period set for calculation and is defined at section Bad Data Max Age **[l]**). The first valid value that is found, started from T-1, is retrieved and takes the place of the bad data (Example of outputs: TAGNAME 10°C BAD)
- Bad Data Max Age **[l]** is used when either HISTORY or MEA Bad Mode are selected **[k]**. It defines the n parameter used in the Bad Data Mode **[k]** and is expressed as a multiplication factor of the aggregation period set for calculation

- Compare Mode **[m]** is the mode selected for handling data comparison and detecting flat tags (i.e. data that are showing a constant value over time). Tags that are showing a same value over the previous run is specified as FLAT 1 while tags that are showing a same value over the previous n run is specified as FLAT 2 (see Nb of Points to Compare **[n]** for definition of the n parameter)
 - NONE: No comparison mode is activated
 - HISTORY: The acquired measurement data at time T are compared with the ones coming from the database at time T-1 to T-n (n is expressed as a multiplication factor of the aggregation period set for calculation and is defined at section Nb of Points to Compare **[n]**)
 - MEA: The acquired measurement data at time T are compared with the ones coming from the backup directory at time T-1 to T-n (n is expressed as a multiplication factor of the aggregation period set for calculation and is defined at section Nb of Points to Compare **[n]**)
- Nb of Points to Compare **[n]** is used when either HISTORY or MEA Compare Mode are selected **[m]**. It defines the n parameter used in the Compare Mode **[m]** and is expressed as a multiplication factor of the aggregation period set for calculation
- Max Flat **[o]** is the allowed percentage of flat tags (expressed in percentage). The percentage is computed based on valid values only (i.e. bad values are not taken into account). If the allowed percentage is exceeded then an action will be taken. The action is defined at section Flat Mode **[p]**
- Flat Mode **[p]** defines the action to be taken in the case allowed percentage **[o]** of flat measurements is exceeded
 - FAILED: an ERF (error file) is generated and the ValiLink action status is failed
 - OK: no specific action is taken and the ValiLink action status is considered as succeeded
 - SERVER: ValiLink action is restarted and ALL tags are acquired on a secondary server. This action is useful in case flat measurements are due to a replica problem in between databases, i.e. the database from which we read data is not synchronized with the database connected to the installation
- Dictionary **[q]** defines the name and path of the Dictionary file which contains the keywords used for digital tags in the external system. Note that the path is relative to the ValiLink configuration file

Example of a dictionary.mea file:

```
OPEN "1"  
OPEN "Ouvert"  
OPEN "Open"  
CLOSED "0"  
CLOSED "Ferme"  
CLOSED "Closed"
```

The objective is to either determine the time fraction of a given time period or the time duration (in seconds), over which a digital tag is OPEN.

In case a dictionary file is referenced, the parameters for either time fraction or time duration calculation are defined through the Actions section **[r]** (Fields [\[43\]](#) [\[49\]](#) [\[50\]](#) [\[53\]](#)).

Attention: this request is only applicable for PI plugs.

The **Actions [r]** allows editing/creating/deleting actions to be carried out, related to the reading/writing from/to external systems. Please see the Actions section for further information.

The **Log File Settings** contains the settings of the logging system. This section is not mandatory. Default data will be used in case fields are left blank.

- Path **[t]** refers to the path where the ValiLink log file is to be stored. By default, the path is the same as the ValiLink.INI file
- File Name **[u]** gives the name of the ValiLink log file. If the field is left blank, the default name is ValiLink.log
- Error level to log **[v]** is used to filter the level of information and error messages to be written in the log file. The log level is to be chosen amongst: NONE, INF, WRN, ERR, ERS and ERF. Messages of a level higher than the one chosen for the log file are written. For instance, if the Error level to log is ERR, the ERR, ERS and ERF messages will be printed only. If the field is left blank, the level INF is used by default
- Backup Option **[w]**
 - BACKUP: the old file ValiLink.log is renamed as ValiLink_bck.log and messages of the current run are logged into a new file ValiLink.log. If the field is left blank, BACKUP is used by default
 - ARCHIVE: messages are written in a new file Valilink_yyyymmdd_hhmmss.log, where yyyymmdd_hhmmss is the current date
 - APPEND: messages of the current run are appended to the old file ValiLink.log
- Maximum Size **[x]** refers to the maximum allowed size of the log file. If the field is left blank, the default size is 10 Mb

3.2.1 BAD DATA HANDLING

A measurement data is defined as Bad when its value is not available at the level of the external system or when its value has no physical meaning such as -9999,999.

Bad Data Mode is intended for applications where retrieving and re-using the last valid measurement values of current faulty sensors is preferable to using default values.

The latest valid measurement values present in the database or backed-up files will be retrieved and used in the reconciliation step, but marked as BAD in the log and results files.

Bad data mode parameters are defined at fields [\[k\]](#) and [\[l\]](#) of the main page of the ValiLink Builder.

3.2.2 FLAT DATA DETECTION

A measurement data is defined as Flat when its value is the same as the n previous runs (where n is an integer set through the 'Nb of points to compare' field [\[n\]](#)). Flat measurements can be due to either frozen sensors or database connection losses, etc.

Two kinds of Flat measurements are distinguished, namely FLAT 1 and FLAT 2:

- FLAT 1 is related to measurement data for which the value is the same as the previous run (n=1)
- FLAT 2 is related to measurement data for which the value is the same as the n consecutive previous runs, where the n parameter is defined by the User (see 'Nb of points to compare' field [\[n\]](#))

Tag_1 reading - Nb of Points to Compare = 3

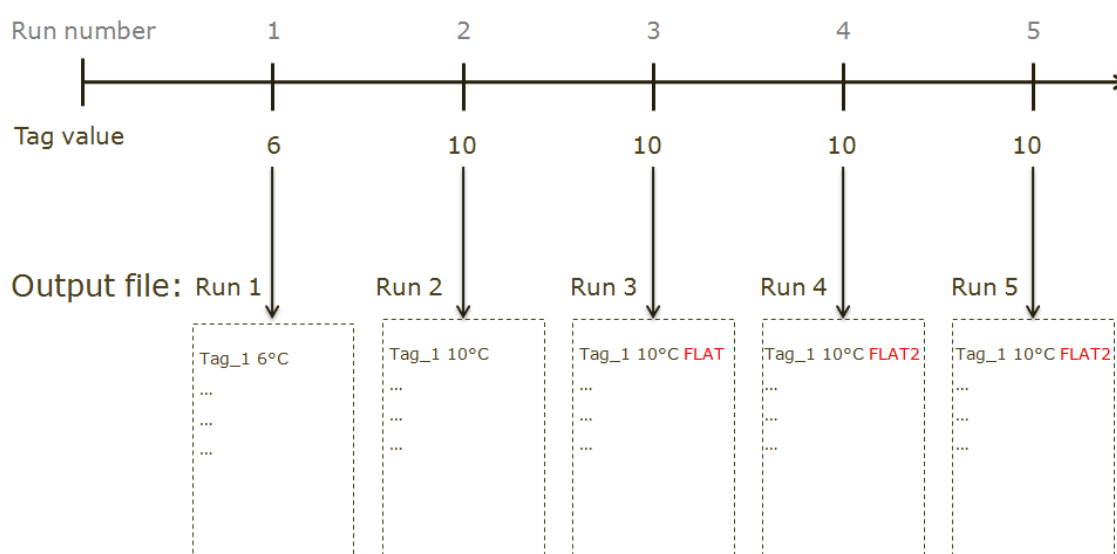


FIGURE 5 – FLAT MODE ILLUSTRATION (FLAT 1 & FLAT 2)

Flat Data detection over the n previous runs (n is set at 'Nb of points to compare' field [\[o\]](#)) is intended to detect flat measurement and taking an action (action is set at 'Flat mode' field [\[p\]](#)) in case the percentage of flat data has exceeded the allowable limit (limit set at 'Max Flat' field [\[o\]](#)). Flat data mode parameters are defined at fields [\[m\]](#), [\[n\]](#), [\[o\]](#) and [\[p\]](#) of the main page of the ValiLink Builder.

Flat data are indicated by the following syntax in the ValiLink output file:

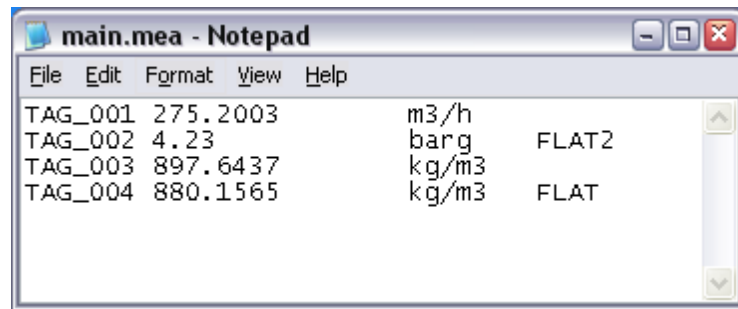


FIGURE 6 – FLAT MODE SYNTAX IN MEA FILE

Note that the flat data syntax is handled by the ValiEngine as if it was a regular tag.

Attention:

- Flat mode should not be used for binary tags as those tags can have a same value (e.g. OPEN) over time
- Flat mode is not applicable to the file plug

3.3 READING AND WRITING ACTIONS

Actions section aims at describing how to carry out the reading/writing from/to the external systems. The actions are defined via the ValiLink Action graphical interface. Several actions can be defined, each ones having a unique number (Action i where i is a unique incremental number). Note that the actions will be performed successively according to the order in which they appear in the ValiLink.INI file and not according to their name numbering i.

To edit/create/delete an action, press the Edit/Add/Remove button of the **Actions window [r]** respectively:

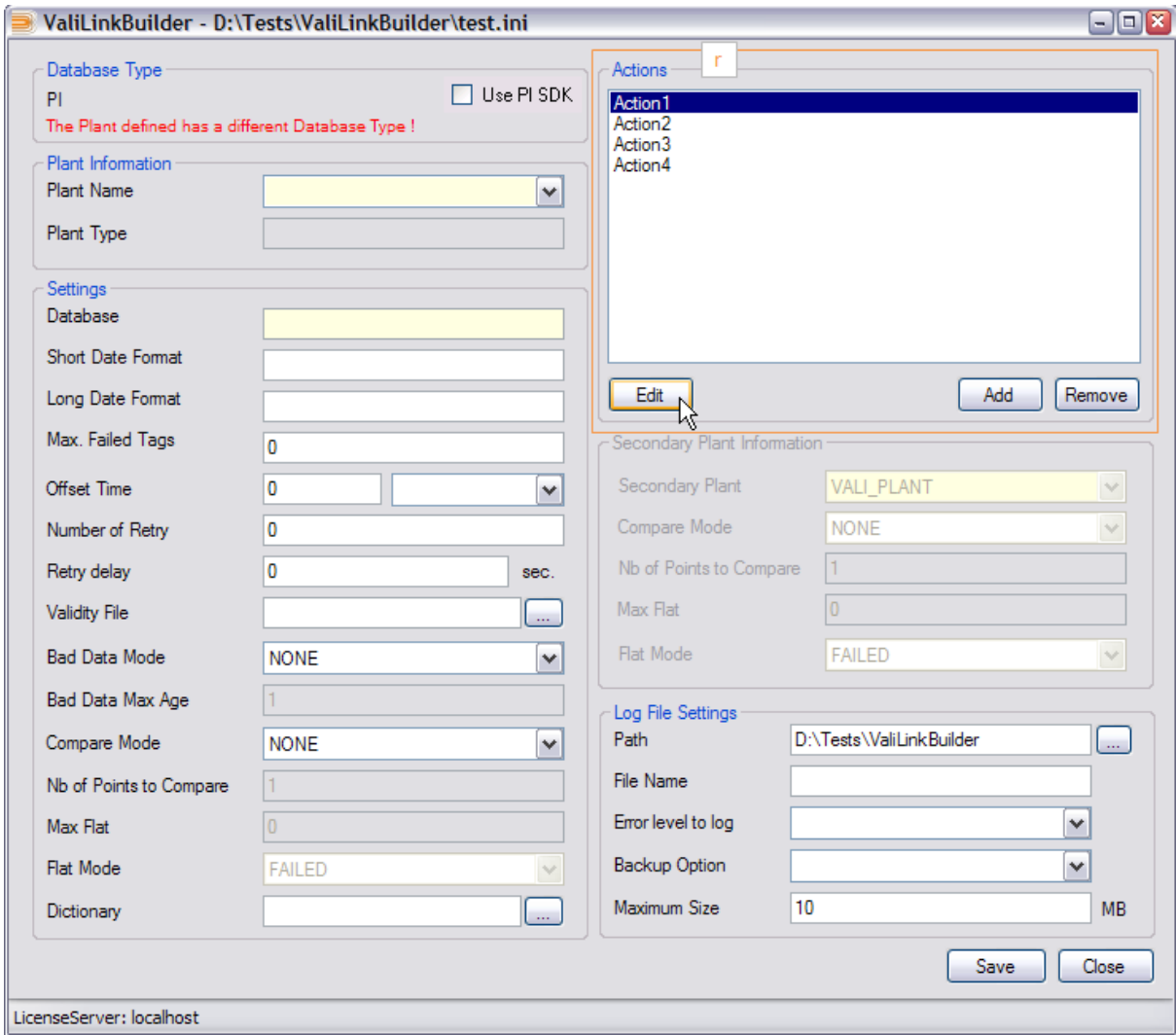


FIGURE 7 – VALILINKBUILDER MAIN WINDOW: ACTIONS

Depending on the database type (defined in Database Type), the ValiLink Action window is automatically adapted.

The description of the fields of the ValiLink Action window are given below. A number is associated to each field, and their description can be found in the recapitulative table (see [Fields Recapitulative Table](#)). The numbering legend is

1	Red numbering	Mandatory field for reading in external system actions
1	Blue numbering	Optional field
	Yellow font	Mandatory field for writing in external system actions

FIGURE 8 – VALILINK BUILDER FIELDS LEGEND

3.3.1 FILE PLUG

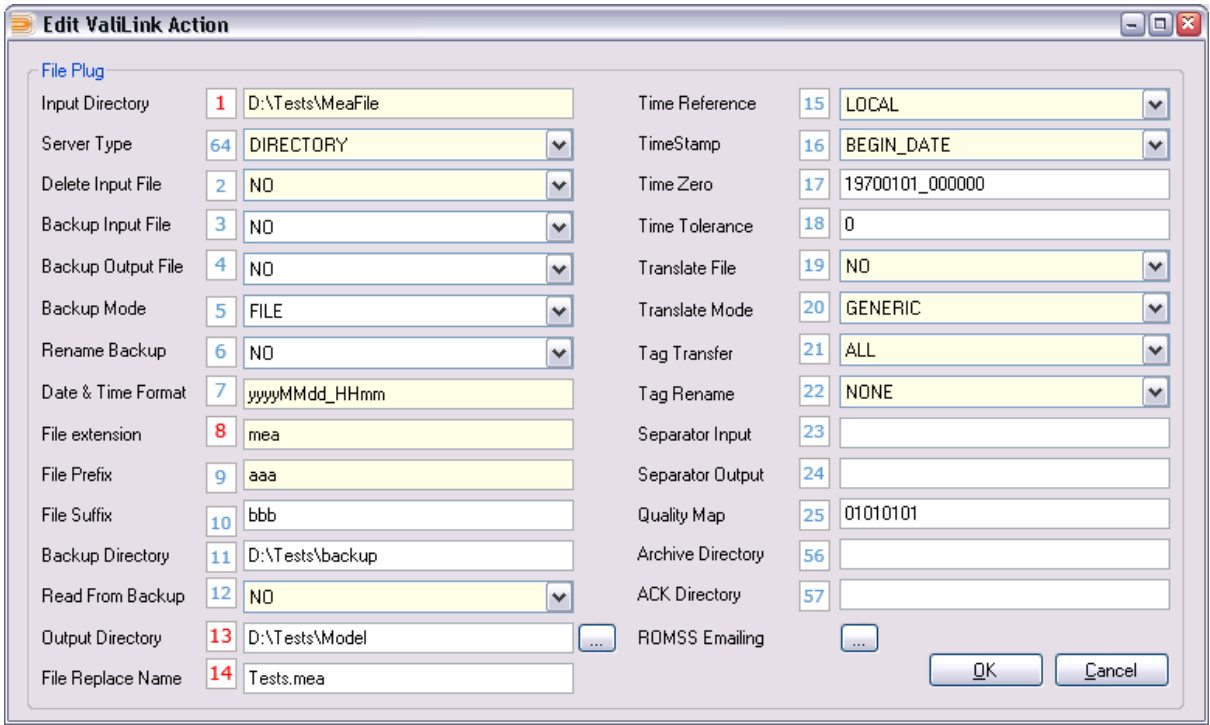


FIGURE 9 – EDIT VALILINK ACTION WINDOW FOR FILE PLUG

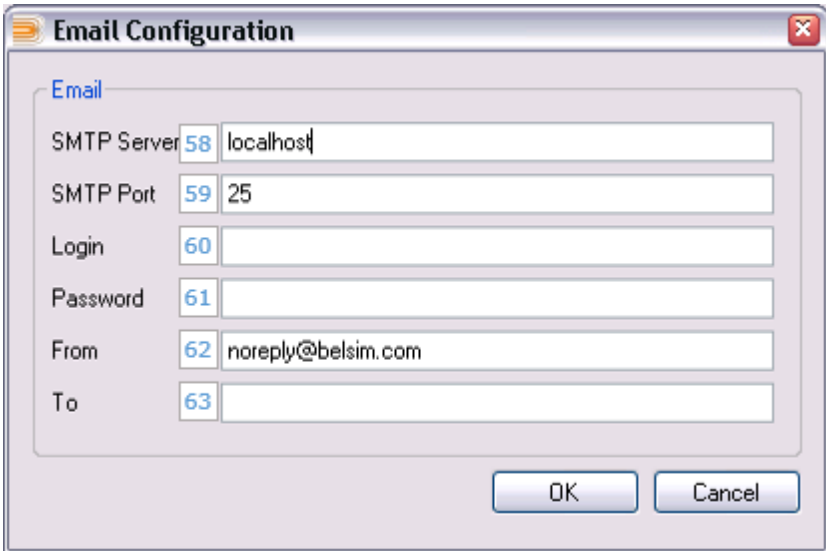


FIGURE 10 – EDIT ROMSS EMAIL CONFIGURATION WINDOW FOR FILE PLUG

The FILE plug can only be configured to read data, as described in [following section](#).

READING DATA FROM FILES

The FILE plug differentiates from other plugs in many ways. It is to be considered as a tool to copy files from one working directory (possibly located on an external server) [1] to another working

directory [\[13\]](#) rather than getting or writing tag values individually. The plug also enables interpreting the contents [\[19\]](#) of the source file or backing-up the copied file [\[3\]](#) in a separate folder [\[11\]](#).

For the specified Begin_Date, ValiLink will search a source folder for a file named according to the following scheme:

FilePrefixYYYYMMDD_HHMMSSFileSuffix.FileExtension

where:

- YYYYMMDD_HHMMSS is the corresponding formatted Begin-Date
- FilePrefix [\[9\]](#), FileSuffix [\[10\]](#) and FileExtension [\[8\]](#)

If the file is found, ValiLink will copy the corresponding file in the output directory [\[13\]](#).

The file can be renamed via the FileReplaceName parameter [\[14\]](#).

Please refer to **Fields recapitulative table** for further information on all fields.

3.3.2 PI, PLANTCONNECT AND OPC/HDA PLUGS

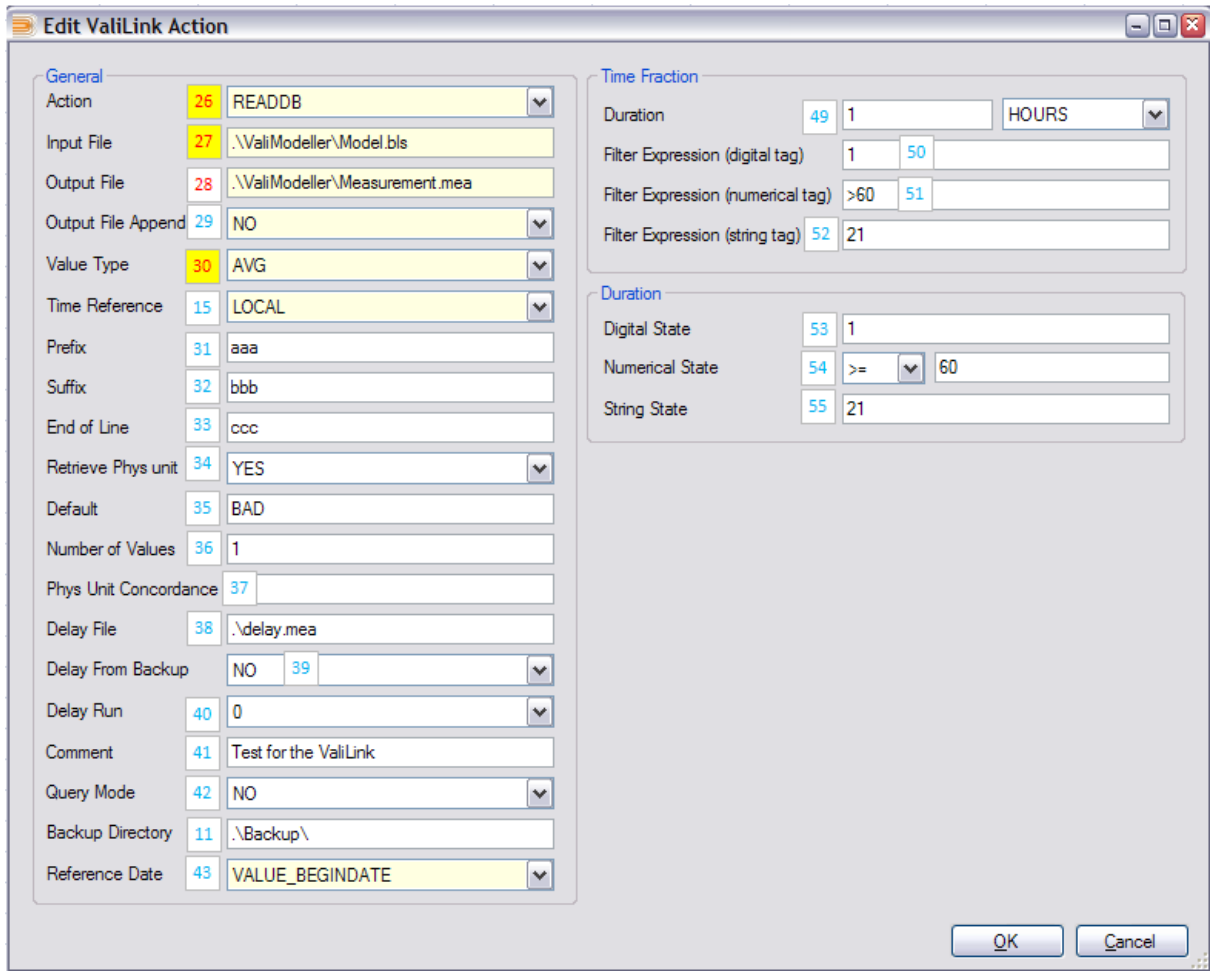


FIGURE 11 – EDIT VALILINK ACTION WINDOW FOR PI, PLANTCONNECT AND OPC HDA PLUGS

The PI, PLANTCONNECT and OPC/HDA plugs can be configured to

- read data from a database
- write data to a database

READING DATA FROM A DIRECT ACCESS DATABASE

The action aims at acquiring from an external system, N values [36] (set by default at 1) of type defined at 'Value Type' field [30] between **BeginDate** and **EndDate** time, for each tags listed in the **InputFile** [27].

The input file [27] can be either a model file or a text file. In case a model file is used, the type of values to be read are to be defined at Tags...Historian Tab of the ValiModeller:

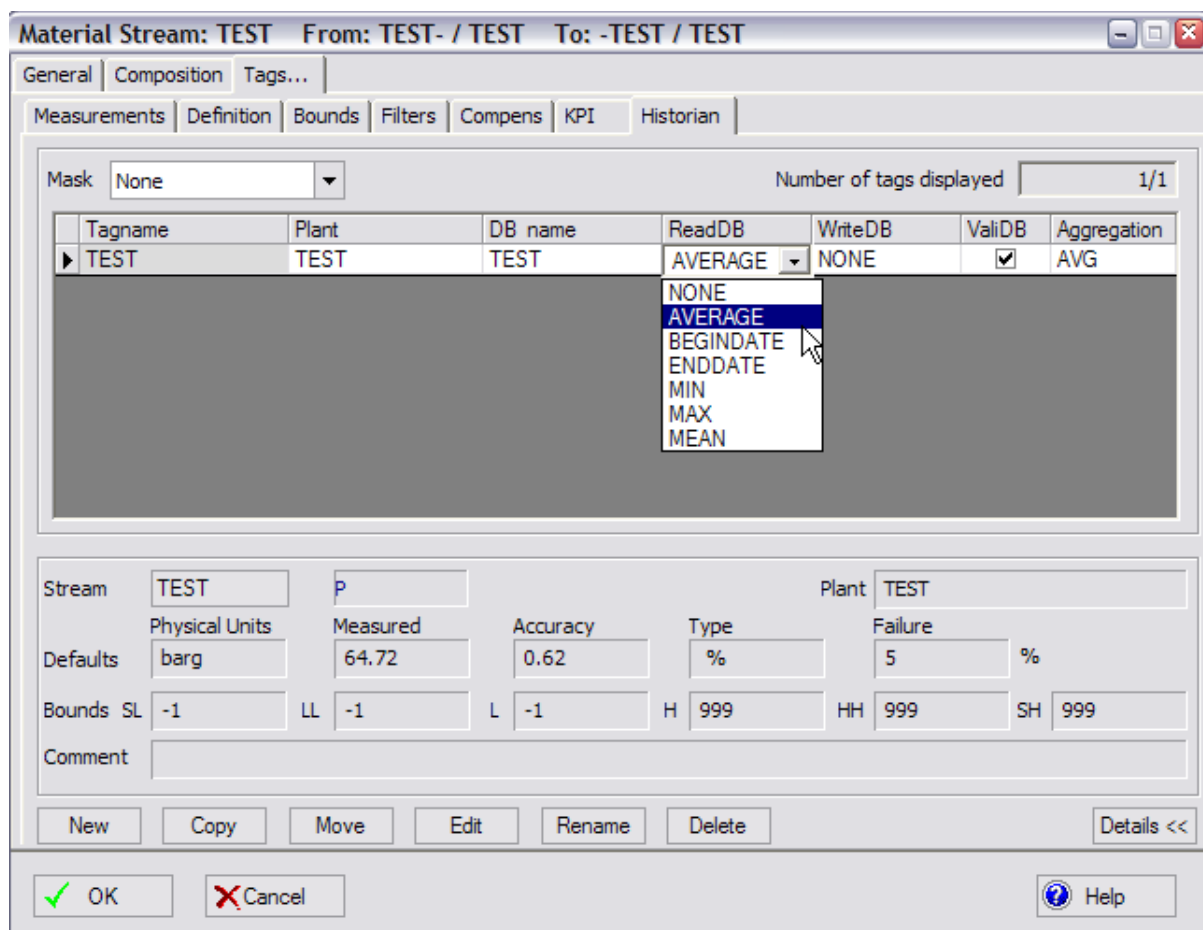


FIGURE 12 - USING MODEL AS INPUT FILE-TYPE OF VALUE DEFINITION

Tags are classified by Plant Name and by Value Type (ReadDB: BEGINDATE, AVERAGE, etc.). Consequently, ValiLink will retrieve amongst the tags defined in the model file, only the ones that correspond to the Value Type [\[30\]](#) defined in the actions.

The results are written in the **OutputFile** [\[28\]](#).

MANDATORY FIELDS

- Action [\[26\]](#): READDDB
- Input File [\[27\]](#)
- Output file [\[28\]](#)
- Value Type [\[30\]](#)

Attention: In case several types of aggregation mode are used, an action should be defined per aggregation mode.

OPTIONAL FIELDS

- Prefix [\[31\]](#)
- Suffix [\[32\]](#)
- EndofLine [\[33\]](#)
- Retrieve Phys unit [\[34\]](#)
- Default [\[35\]](#)
- Number of Values [\[36\]](#)
- DataBase [\[45\]](#)
- Delay File [\[38\]](#)
- Delay From Backup [\[39\]](#)
 - Delay Run [\[40\]](#)
 - Comment [\[41\]](#)
 - Backup Directory [\[11\]](#)

Please refer to **Fields recapitulative table** for further information on all fields.

WRITING DATA TO A DIRECT ACCESS DATABASE

The action aims at writing data to an external system for each tags specified in the **InputFile** [\[27\]](#).

MANDATORY FIELDS

- Action [\[26\]](#): WRITEDB
- InputFile [\[27\]](#)
 - Value Type [\[30\]](#)

OPTIONAL FIELDS

- Prefix [\[31\]](#)
- Suffix [\[32\]](#)
- DataBase [\[45\]](#)

ADDITIONAL INFORMATION

Additional information can be configured:

- time fraction
- duration calculation

TIME FRACTION

ValiLink can detect the time fraction during which a given condition is satisfied for a given tag and over a given period.

For instance, if a digital tag indicates that a piece of equipment is functioning or not, ValiLink can detect the time fraction over which this unit has been stopped during a specific time span (e.g. 5 hours before the present run).

This allows taking preventive actions if a transient behavior is expected after the restarting of a piece of equipment (example: well restart).

ValiLink will output in a mea file, a value in between 0 and 1 associated to a new tag, composed of the original tag name and the suffix “_TIMEFRAC”. Because of the variation range of those values [0;1], those tags can directly be used as status tags in ValiModeller.

The condition to be satisfied can be either related to digital, analogical or string tags. In case of a digital tag, the conditions are 0 or 1 (Note that the mapping to explicit keywords such as OPEN, CLOSED, etc. is done through the dictionary file [ql]).

Attention: For activating the time fraction calculation, the Value Type field [30] has to be defined to “Time_fraction”.

DURATION CALCULATION

For a specific tag, ValiLink can evaluate the time duration for which a given condition is satisfied.

For instance, a digital tag has 2 values: 0 or 1, meaning that a piece of equipment is functioning or not respectively. At the current run, the unit is functioning. ValiLink can detect the time duration over which the piece of equipment has been active.

The output will be written in a mea file and is expressed in seconds. It will be associated with a tag composed of the original tag name and the suffix “_DURATION”.

Attention: For activating this feature, the Value Type field [30] has to be defined to “Duration”.

Please refer to **Fields recapitulative table** for further information on all fields.

3.3.3 SQL, ACCESS, IP21, ORACLE and ODBC plugs

FIGURE 13 – EDIT VALILINK ACTION WINDOW FOR SQL, ACCESS, ORACLE, IP21 AND ODBC PLUGS

The SQL plug can be configured to

- read data from a database
- write data to a database

READING DATA FROM A MS-SQL DATABASE

The action aims at acquiring from an external system, **NValue** [36] values (by default 1) for each of the tags specified in the **InputFile** [27], through the execution of a **SELECT** query [46]. The results are written in the **OutputFile** [28].

MANDATORY FIELDS

- Action [26]: READDB
- InputFile [27]
- OutputFile [28]
- StrSQL_SELECT [46]

OPTIONAL FIELDS

- Prefix [\[31\]](#)
- Suffix [\[32\]](#)
- EndofLine [\[33\]](#)
- Default [\[35\]](#)
- DataBase [\[45\]](#)

Please refer to **Fields recapitulative table** for further information on all fields.

WRITING DATA TO A MS-SQL DATABASE

The action aims at writing data to an external system for each tags specified in the **InputFile** [\[27\]](#) through the execution of an INSERT [\[47\]](#) and/or UPDATE [\[48\]](#) query.

MANDATORY FIELDS

- Action [\[26\]](#): WRITEDB
- InputFile [\[27\]](#)
- Query Order [\[44\]](#)
- StrSQL_INSERT [\[47\]](#)
- StrSQL_UPDATE [\[48\]](#)

OPTIONAL FIELDS

- Prefix [\[31\]](#)
- Suffix [\[32\]](#)
- QueryOrder [\[44\]](#)
- DataBase [\[45\]](#)

Please refer to **Fields recapitulative table** for further information on all fields.

3.4 FIELDS RECAPITULATIVE TABLE

Description of the fields is given below:

FIELD	FIELD NAME	FORMAT	DESCRIPTION
[1]	Input Directory	String	Specify the directory where the input source files/folders to be copied are stored

FIELD	FIELD NAME	FORMAT	DESCRIPTION
[2]	Delete Input File	[YES NO]	Choose to delete or not the input source files/folders once copied or interpreted
[3]	Backup Input File	[YES NO]	Choose to make a backup or not of the input source files/folders being copied or interpreted
[4]	Backup Output File	[YES NO]	Choose to make a backup or not of the output files/folders
[5]	Backup Mode	[FILE FOLDER]	Define to either backup the files in the backup folder directly [11] (FILE) or in a sub-folder of the backup folder for each input source file (FOLDER)
[6]	Rename Backup	[YES NO]	Choose to rename or not the backup files. The file replace name is the one defined at field [14]
[7]	Date & Time Format	String	Specify the date and time format used to determine the name of the input source files/folders to be copied or interpreted
[8]	File extension	String	Specify the extension of the input source file to be copied or interpreted
[9]	File Prefix	String	Specify the strings to be added before a given date for the input file name search. ValiLink will search a source folder for a file, named according to the following scheme: FilePrefixyyyymmdd-hhmmss.FileExtension[8]
[10]	File Suffix	String	Specify the strings to be added after a given date for the input file name search. ValiLink will search a source folder for a file, named according to the following scheme: FileSuffixyyyymmdd-hhmmss.FileExtension[8]
[11]	Backup Directory	String	Directory path where input [3] and/or output [4] files are backed-up
[12]	Read From Backup	[YES NO]	In case the input files cannot be found in the Input Directory [1], use the corresponding one

FIELD	FIELD NAME	FORMAT	DESCRIPTION
			in Backup Directory [11] if any
[13]	Output Directory	String	Output Directory path where input files are copied (destination folder)
[14]	File Replace Name	String	Specify the name to be given to the input files that are copied in the Output Directory [13]
[15]	Time Reference	[LOCAL UTC]	Specify the time to be used as a reference for determining the date and time: - LOCAL use local time as the reference to determine the date time - UTC use UTC time as the reference to determine the date time
[16]	TimeStamp	[BEGIN_DATE END_DATE BEGIN_TIME END_TIME]	Specify if the date and time used for the input files are the begin or end date and time: - BEGIN_DATE use the begin date to determine the date time - END_DATE use the end date to determine the date time - BEGIN_TIME use the begin date time to determine the date time - END_TIME use the end date time to determine the date time
[17]	Time Zero	String (yyyyMMdd_HHmss)	When working with date and time expressed in number format, the Time Zero gives the date and time corresponding to zero number. If left blank, the default Time Zero is 01/01/1970. The Zero date time is used to specify the date time for determining the name of the input file/folder
[18]	Time Tolerance	String	Tolerance in seconds allowed for the name of the input files. This option is usefull in case data bases cannot generate files with exact dates and times

FIELD	FIELD NAME	FORMAT	DESCRIPTION
[19]	Translate File	[YES NO]	Specify if the input files are to be either copied or interpreted
[20]	Translate Mode	[GENERIC WINIS]	Specify the translation mode: ▪ GENERIC Generic mode of translation ▪ WINIS Mode of translation specific to WinIS
[21]	Tag Transfer	[ALL LICENSESERVER]	In case files are interpreted [19], it specifies which tags should be interpreted: either all tags defined in the input file (ALL) or tags defined in the LicenseServer only (LICENSESERVER)
[22]	Tag Rename	[NONE DBNAME]	Choose to rename or not the tags with the corresponding DB name
[23]	Separator Input	String	Specify the character used as field separator in the input files (TAB/;/space/etc.)
[24]	Separator Output	String	Specify the character used as field separator in the output files (Only TAB and Space are recognized by the Engine). If the field is left blank, the space is used as the separator by default
[25]	Quality Map	String Quality Map = 01010101	WinIS has 7 possible codes to determine the quality of the values provided by the system (byte values): 1 Value OK 2 Value failed 3 Failed with substitute value 4 Value hidden 5 Hidden with substitute value 6 Value defined but not available
[26]	Action	[READDB WRITEDB]	Specify the type of actions to be performed: read data from database or write data to

FIELD	FIELD NAME	FORMAT	DESCRIPTION
			database
[27]	Input File	String	Specify the path of and name of the file that contains the tags to be transferred (model or text files) - One tag per line
[28]	Output File	String	Specify the path and name of the output file to be written by ValiLink
[29]	Output File Append	[YES NO]	In case of several actions, choose to either append the new values at the end of the output files or to clear the files before writing
[30]	Value Type	[AVG MEAN MIN MAX VALUE_BEGINDATE VALUE_ENDDATE INTEGRAL DURATION TIME_FRACTION STDDEV]	Specify the type of values to be transferred:

FIELD	FIELD NAME	FORMAT	DESCRIPTION
			■ AVG (for READDDB action only [26]) read average value for the defined time period (if relevant, a time weighted value will be provided) ■ MEAN (for READDDB action only [26]) read arithmetical mean of values available for the defined time period ■ MIN (for READDDB action only [26]) read minimum value for the defined time period ■ MAX (for READDDB action only [26]) read maximum value for the defined time period ■ VALUE_BEGINDATE (for READDDB and WRITEDDB action [26]) read instant value for the beginning of the defined time period / write value with the time stamp of the beginning of the time period ■ VALUE_ENDDATE (for READDDB and WRITEDDB action [26]) read instant value for the end of the defined time period / write value with the time stamp of the end of the time period ■ DURATION (for READDDB action only [26]) determine the time duration over which a given condition is satisfied. The condition and reference date are to be defined at fields [43] and [53], [54] or [55] ■ TIME FRACTION (for READDDB action only [26]) determine the time fraction of a given time span, over which a given condition is satisfied. The condition and reference date are to be defined at fields [43], [49] and [50], [51] or [52] ■ STDDEV (for READDDB action only [26]) read the standard deviation for the defined time period. Note that:

FIELD	FIELD NAME	FORMAT	DESCRIPTION
			▪ The input file [27] (with the tag list) has to be a text file (not a model file). ▪ STDDEV is only available with PI (OSIsoft ©).
[31]	Prefix	String	Specify the strings to be written before the name of the tags in the output files / external system
[32]	Suffix	String	Specify the strings to be written after the name of the tags in the output files / external system
[33]	End of Line	String	Specify the strings to be written at the end of each line in the output files. Usually the string will be enclosed by double quotes " and start by a space (Example: ENDOFLINE=" T/HR")
[34]	Retrieve Phys Unit	[YES NO]	Specify if physical units associated to a tag are to be read in the external system and written it in the output file or not
[35]	Default	Real number or string	Specify the values to be written in case no value is retrieved from the external system. If left blank, BAD is used by default. Setting the value to BAD enables to activate a corrective action by ValiFilter and to trace the acquisition issue in ValiEngine html reports
[36]	Number of Values	Positive integer	Specify the number of values to be read for each tag (by default 1). The time window will be divided in Nvalue intervals and a value will be produced for each interval
[37]	Phys Unit Concordance	Not used	Not used
[38]	Delay File	String	Specify the name and path of the delay file which is containing the delays to be applied for each zone (one zone per line, syntax of each line: ZONE_DELAY 100.012)

FIELD	FIELD NAME	FORMAT	DESCRIPTION
			The ZONE_DELAY are to be defined in the model.bls file. Applicable only in case the input file [27] is a model file If no physical unit is specified, seconds will be used by default
[39]	Delay From Backup	[YES NO]	Specify if the delay file is to be read in the backup directory [11] or not: ▪ YES Delay file is read from Backup directory ▪ NO Delay file is not read from backup directory
[40]	Delay Run	[0 -1]	Specify if the delay file to be read is the one of the current run (0) or the one of the previous run (-1)
[41]	Comment	string	Provide a description to Action i The description will be displayed in the Action section of the ValiLinkBuilder main window [r]
[42]	Query Mode	[YES NO]	Applicable for reading actions [26] in SQL database only. Enable User's query definition. If YES, the query is to be defined at StrSQL_SELECT [46]
[43]	Reference Date	[VALUE_BEGINDATE VALUE_ENDDATE]	Specify the reference date to be used as the starting point to calculate time fraction and duration
[44]	Query Order	[INSERT UPDATE]	Specify the query to be performed first in between INSERT and UPDATE. By default, INSERT is performed first ▪ INSERT: StrSQL_INSERT query is attempted first ▪ UPDATE: StrSQL_UPDATE query is attempted first

FIELD	FIELD NAME	FORMAT	DESCRIPTION
[45]	Database	String	Specify the name of the database to connect to for the action in case it is different from the one defined in the plant (note that it overwrites the SERVER SETTINGS section). In case of an Access database, the path and name of the Database should be provided
[46]	StrSQL_SELECT	String	SQL query for SELECT command to be executed to get the value of the tag. This query can refer to pre-defined keywords [KeyWords]
[47]	StrSQL_INSERT	String	SQL query for INSERT command to insert a new record in the external system
[48]	StrSQL_UPDATE	String	SQL query for UPDATE command to update an existing record in an external system. By default (see QueryOrder [44]), an INSERT is attempted first. If it fails than an UPDATE is carried out in place. This query can refer to pre-defined keywords [KeyWords]
Keywords			Following keywords can be used:

FIELD	FIELD NAME	FORMAT	DESCRIPTION
			▪ %BeginDateShort%: date information related to BeginDate ▪ %BeginDateLong%: date and time information related to BeginDate ▪ %EndDateShort%: date information related to EndDate ▪ %EndDateLong%: date and time information related to EndDate ▪ %Hour%: number of hours between BeginDate and EndDate ▪ %INPUT1%: first column from file InputFile ▪ %INPUT2%: second column from file InputFile ▪ %INPUT3%: third column from file InputFile
[49]	Time fraction - Duration	Positive integer	Specify the time to be subtracted to the reference date [43] to obtain a time period. The time period will be used to determine the time fraction over which a condition is satisfied [50] [51] [52]. Option available for PI only
[50]	Time fraction - Filter Expression (digital tag)	Digital	Used to specify the condition to be satisfied for the Time Fraction determination in case of digital tag
[51]	Time fraction - Filter Expression (numerical tag)	Numerical	Used to specify the condition to be satisfied for the Time Fraction determination in case of numerical tag - legal operators to be used are <, >, =, <=, >=
[52]	Time fraction - Filter Expression (string tag)	String	Used to specify the condition to be satisfied for the Time Fraction determination in case of string tag
[53]	Duration - Digital State	Digital	Used to specify the condition to be satisfied for the Duration determination in case of digital tag

FIELD	FIELD NAME	FORMAT	DESCRIPTION
[54]	Duration Numerical State	Numerical	Used to specify the condition to be satisfied for the Duration determination in case of numerical tag
[55]	Duration - String State	String	Used to specify the condition to be satisfied for the Duration determination in case of string tag
[56]	Archive Directory	String	Used to specify the FTP Archive directory used with XML_ROMSS Translate mode
[57]	ACK Directory	String	Used to specify the FTP Acknowledgment directory used with XML_ROMSS Translate mode
[58]	SMTP Server	String	Used to specify the SMTP Server used to send advertisement email with XML_ROMSS Translate mode
[59]	SMTP Port	Positive integer	Used to specify the SMTP Port of the SMTP Server used to send advertisement email with XML_ROMSS Translate mode. Default value is 25
[60]	Email Login	String	Used to specify the Login of the SMTP Server used to send advertisement email with XML_ROMSS Translate mode.
[61]	Email Password	Encrypted string	Used to specify the Password of the SMTP Server used to send advertisement email with XML_ROMSS Translate mode.
[62]	Email From	String	Used to specify the email address of the sender used to send advertisement email with XML_ROMSS Translate mode. Default value is noreply@belsim.com
[63]	Email To	String	Used to specify the email address of the recipients used with email feature of XML_ROMSS Translate mode. ';' character is used as recipients separator.
[64]	Server Type	[FOLDER]	Specify the

FIELD	FIELD NAME	FORMAT	DESCRIPTION
		[FTP]	

SECTION 4: INPUT AND OUTPUT FILES FORMAT

4.1 COMMON FORMATTING CONVENTIONS

REPRESENTATION OF REAL NUMBERS

The standard format use '.' as separation character. Using ',' is however also supported.

Examples:

```
342.524
0.485640E-01
-23813,3
0.244926E+14
```

SEPARATION CHARACTERS

Tab and Space can be used as separation character.

4.2 FORMAT OF THE INPUT FILES

GETTING DATA FROM AN EXTERNAL SYSTEM

The Input File lists the tags for which one or more values are requested from an external system.

The tags must be listed one per line.

Example:

```
PI-001
TI-002
FI-001
...
```

WRITING DATA TO AN EXTERNAL SYSTEM

The Input File contains the list of tags as well as their value. Physical units can be given but are ignored.

Typically the Input File will have been generated by VALI as an OUTPUT measurement file.

Example:

```
PI-001 27.3000      barg
TI-001 192.000      C
FI-001 1003.42      t/h
...
```

4.3 FORMAT OF THE OUTPUT FILES

The tag values acquired from an external system will be written in the Output File in a format directly readable by VALI as measurement file.

Physical units (also known as engineering units) are written in the output file by the application when they are provided by the external system. Should a space character appear in these physical units, it will be replaced by the underscore character: "_". For example, Bar g will be converted into Bar_g.

The three fields (tag name, value and physical units) are separated by a tab.

Example:

```
PI-001 <tab> 27.3000 <tab> barg  
TI-001 <tab> 192.000 <tab> C  
FI-001 <tab> 1003.42 <tab> t/h  
...
```

SECTION 5: CONTENTS OF THE LOG FILE

Error messages are identified by an alphanumeric code of the following form:

[DATE TIME] COD NUM: MESSAGE

where:

- DATE: the date
- TIME: the time:
- COD: the level of severity of the error:
 - **INF**: Information:
 - connection to a server or database
 - successful read or write
 - sql query message
 - **WRN**: Warning:
 - default value is used when a failure occurred on a tag reading
 - **ERR**: Failure at the level of a Tag
 - tag is not defined in server
 - failed to get/write value of a tag
 - **ERS**: Failure at the level of an Action:
 - failed to read input file
 - failed to write output file
 - **ERF**: Failure at the level of the application:
 - failed to read ini file
 - invalid dates format
 - connection to server failed
- NUM: the error number (4 digits)
- MESSAGE: message describing the issue

SECTION 6: PLUGS TECHNICAL INFORMATION

6.1 PI

PI REQUIREMENTS SERVER SIDE

The PI server will be configured to enable reading and receiving data from VALI. In that respect, the adequate number of TAGs will be made available in the PI database.

For further technical details regarding the PI server configuration, please visit OsiSoft support site at support.OSIsoft.com.

REQUIREMENTS CLIENT SIDE

There are two ways of querying the PI database:

- through PI Software Development Kit (SDK),
- or through PI Advanced Programming Interface (API). The PI connector uses the API routines included in the PI-API32.dll library. ValiLink is compliant with PI-API32.dll version 1.3.8.0 and later.

Any of the following separate installation will enable a working connection on the client side:

- the PI-software development kit (PI-SDK version 1.2.0): The PI Software Development Kit (PI-SDK) is a programming tool providing access to PI Servers. The software consists of an ActiveX in-process server, an ActiveX control, and supporting code libraries
- PI ProcessBook (PI-PB version 2.31): A graphics package that allows users to create dynamic, interactive graphical displays featuring real-time PI data

The PI connector must be installed on the computer where VALI software is installed. No further software is needed.

On top of the requirements described above, the customer will enable the connection at the network level: generally, customer network security policies do not interfere with the connection between ValiLink and the PI server.

A working PI connection can be tested thanks to the PI ProcessBook on the computer where VALI is installed.

ValiLink is compliant with PI Trust to simplify the connection with PI DAHS that are managed through PI Trust.

Attention:

1. When reading a value, ValiLink will always retrieve the extrapolated value.
2. The value can be a numeric value, a string or a digital point.

6.2 IP21

IP21 REQUIREMENTS SERVER SIDE

ValiLink is fully compliant with Infoplus.21 4.1.1 or higher. The IP21 server will be configured to enable reading and receiving data from VALI. In that respect, the adequate number of TAGs will be made available in the IP21 database.

For further technical details regarding the IP21 server configuration, please visit AspenTech support site at support.aspentech.com.

IP21 REQUIREMENTS CLIENT SIDE

ValiLink requires the following software installed on client computer for IP21 connection:

- SQLplus ODBC driver (contained in the installation CD-Rom of IP21)

No further software is needed.

On top of the requirements described above, the customer will enable the connection at the network level: generally, customer network security policies do not interfere with the connection between ValiLink and the IP21 server.

ODBC CONFIGURATION

ValiLink uses ODBC connection to provide data access to the IP21 Database.

ODBC (Open Database Connectivity) is a standard database access method. The goal of ODBC is to access data from any application, regardless of which DataBase Management System (DBMS) is handling the data. ODBC manages this by inserting a middle layer, called a database driver, between an application and the DBMS. The purpose of this layer is to translate the application's data queries into commands that the DBMS understands. For this to work, both the application and the DBMS must be ODBC-compliant that is, the application must be capable of issuing ODBC commands and the DBMS must be capable of responding to them.

Before connecting to the Database, the user must first create a DSN (Data Source Name) via the **"ODBC Data Source Administrator"** program found in the client computer's Control Panel (in Administrative Tools menu). Make sure to create a SYSTEM DSN (not a USER DSN).

Click "add" in the menu and choose in the list "AspenTech SQLplus" to create a new data source.

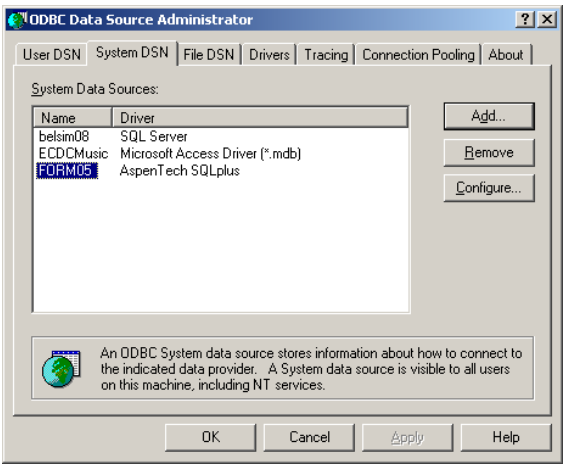


FIGURE 6 - ODBC DATA SOURCE ADMINISTRATOR

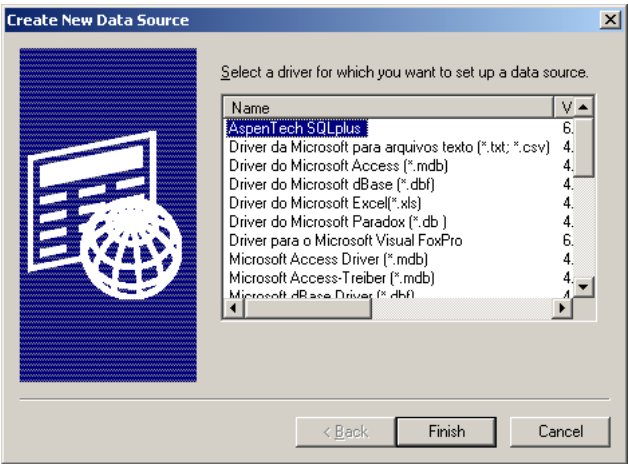


FIGURE 7 - CREATE A NEW OBBC DATA SOURCE

The AspenTech IP21 ODBC connection uses the Windows login to connect to the database. There are no identifying and password to introduce in the configuration.

To configure the data source, provide the following information::

- the name of the data source (user defined ID)
- a short description
- the IP or UNC of the IP21 server
- the TCP/IP port of the server (default is 10014)

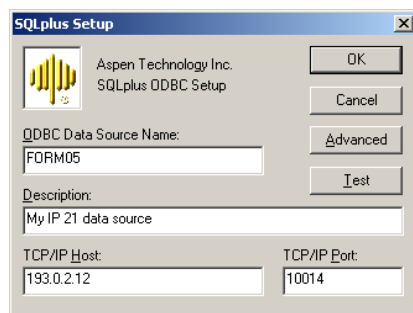


FIGURE 8 - CREATE A NEW ODBC DATA SOURCE

Press the **Test** button to test the connection is up before continuing.

DATE TIME FORMAT

AspenTech IP21 uses a fixed date time format "yyyy-MM-dd HH:mm:ss" which is encoded directly in the ValiLink software. However, in each SQL query, the reserved word **TIMESTAMP** must be specified before date field to ensure correct conversion.

REQUEST WORD

The reserved word "request" is used to specify the type of data returned. By default, the value is 1 but in our case, value must be set to 2 (used for instant value). For more information, please refer to the SQLplus help.

SELECT QUERIES

ValiLink enables to use all the SQL language detailed in the SQLplus help to read and write data to/from IP21 Databases.

Here are some typical queries used in ValiLink:

Get one Value at one time:

```
StrSQL_select=SELECT VALUE FROM HISTORY WHERE NAME = '%INPUT1%' AND
                TS = TIMESTAMP'%BeginDateLong%' AND REQUEST = 2;
```

Get one Value at one time with physical unit:

```
StrSQL_select=SELECT VALUE, (SELECT IP_ENG_UNITS FROM IP_ANALOGDEF
                              WHERE NAME = '%INPUT1%')
                AS PhysUnit FROM HISTORY WHERE NAME = '%INPUT1%'
                AND TS = TIMESTAMP'%BeginDateLong%' AND REQUEST = 2;
```

Average on one Value:

```
StrSQL_select=SELECT AVG, (SELECT IP_ENG_UNITS FROM IP_ANALOGDEF WHERE NAME = '%INPUT1%')
AS PhysUnit FROM AGGREGATES WHERE NAME = '%INPUT1%'
AND TS BETWEEN TIMESTAMP'%BeginDateLong%' AND TIMESTAMP'%EndDateLong%'
AND request = 2
AND PERIOD = (TIMESTAMP'%EndDateLong%' - TIMESTAMP'%BeginDateLong%');
```

Average with time period divided by 10:

```
StrSQL_select=SELECT AVG, (SELECT IP_ENG_UNITS FROM IP_ANALOGDEF WHERE NAME = '%INPUT1%')
AS PhysUnit FROM AGGREGATES WHERE NAME = '%INPUT1%'
AND TS BETWEEN TIMESTAMP'%BeginDateLong%' AND TIMESTAMP'%EndDateLong%'
AND request = 2
AND PERIOD = ((TIMESTAMP'%EndDateLong%' - TIMESTAMP'%BeginDateLong%')/10);
```

Maximum on one value:

```
StrSQL_select=SELECT MAX, (SELECT IP_ENG_UNITS FROM IP_ANALOGDEF WHERE NAME = '%INPUT1%')
AS PhysUnit FROM AGGREGATES WHERE NAME = '%INPUT1%'
AND TS BETWEEN TIMESTAMP'%BeginDateLong%' AND TIMESTAMP'%EndDateLong%'
AND request = 2
AND PERIOD = (TIMESTAMP'%EndDateLong%' - TIMESTAMP'%BeginDateLong%');
```

Minimum on one value:

```
StrSQL_select=SELECT MIN, (SELECT IP_ENG_UNITS FROM IP_ANALOGDEF WHERE NAME = '%INPUT1%')
AS PhysUnit FROM AGGREGATES WHERE NAME = '%INPUT1%'
AND TS BETWEEN TIMESTAMP'%BeginDateLong%' AND TIMESTAMP'%EndDateLong%'
AND request = 2
AND PERIOD = (TIMESTAMP'%EndDateLong%' - TIMESTAMP'%BeginDateLong%');
```

INSERT-UPDATE QUERIES

The INSERT commands always insert a new value, even if there is already a record with same data. It is therefore recommended to always use the UPDATE command first.

Attention:

1. The primary key is fixed on an ID, not on the date time!
2. It is not possible to delete a record from the database.

Insert a new Value:

```
StrSQL_insert=INSERT INTO "%INPUT1%" (IP_TREND_VALUE,IP_TREND_TIME)
VALUES (%INPUT2%,TIMESTAMP'%BeginDateLong%');
```

Update a Value:

```
StrSQL_update=UPDATE "%INPUT1%" SET IP_TREND_VALUE = %INPUT2%
WHERE IP_TREND_TIME = TIMESTAMP'%BeginDateLong%';
```

By default, it is only possible to write a value with a time stamp falling within the 24 hours period before execution time. This setting can be modified using the Server application of IP21:

- right click in the historical table (under "historian" leaf).
- see IP_REPOSITORY in the field structure IP_ANALOG_DEF of the tag.
- choose "Properties" and change time value
- restart the database must be restarted for changes to take effect

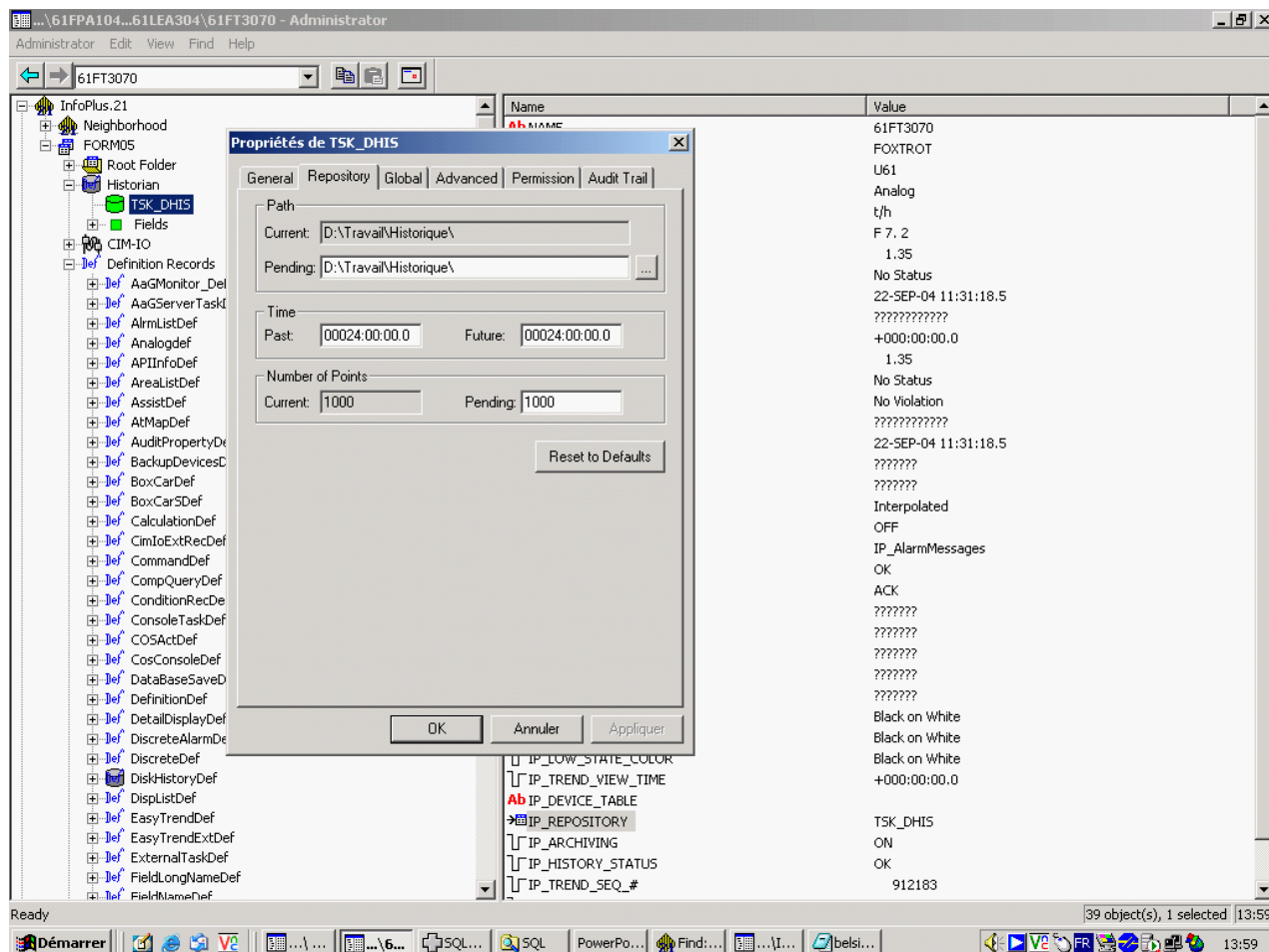


FIGURE 9 - CHANGING PROPERTIES OF IP21 SERVER

EXAMPLE OF VALILINK INI FILE FOR IP21

```
[SERVER SETTINGS]
PlantName=CP_IP21
DataBase=FORM05
ShortDateFormat=yyyy-MM-dd
LongDateFormat=yyyy-MM-dd HH:mm:ss

[LOGFILE OPTION]
Path=
LogLevel=0
Backup Option=BACKUP

[Action1]
InputFile=ToSQL_all.txt
```

```
OutputFile=FromSQL.mea
StrSQL_select=SELECT VALUE, Phys_Unit FROM db WHERE NAME = '%INPUT1%' AND Date =
'%BeginDateLong%';

[Action2]
QueryOrder=update
InputFile=SQLtest.out
StrSQL_update=Update "%INPUT1%" SET IP_TREND_VALUE = %INPUT2% WHERE IP_TREND_TIME =
TIMESTAMP'%EndDateLong%';
StrSQL_insert=INSERT INTO "%INPUT1%" (IP_TREND_VALUE, IP_TREND_TIME) VALUES
(%INPUT2%, TIMESTAMP'%BeginDateLong%');
```

6.3 PLANTCONNECT

PLANTCONNECT REQUIREMENTS SERVER SIDE

ValiLink is fully compliant with PlantConnect 3.5 or higher

The PlantConnect server must be configured to enable reading and receiving data from VALI. A dedicated user with the appropriate rights must be created to enable the access to PlantConnect.

For further technical details regarding the PlantConnect server configuration, please contact ABB technical support.

PLANTCONNECT REQUIREMENTS CLIENT SIDE

ValiLink requires the following software installed on client computer for PlantConnect connection:

- PlantConnect client installation 3.5

On top of the requirements described above, the customer must enable the connection at the network level: generally, customer network security policies do not interfere with the connection between ValiLink and PlantConnect.

6.4 MS-SQL

MS-SQL REQUIREMENTS SERVER SIDE

ValiLink is fully compliant with MS-SQL 7.0 or higher

The MS-SQL server must be configured to enable reading and receiving data from VALI. A dedicated user with the appropriate rights must be created to enable the access to the MS-SQL database.

For further technical details regarding the Oracle server configuration, please visit Microsoft support site at <http://www.microsoft.com/sql/support/default.msp>.

MS-SQL REQUIREMENTS CLIENT SIDE

ValiLink can connect to MS-SQL using 3 standard connection plugs:

- OLEDB
- ODBC
- .NET native driver

The .Net native driver gives the best performance.

ValiLink requires the following software installed on client computer for MS-SQL connection:

- .NET framework 2.0 (included in VALI4 installation package)

The .NET framework must be installed on the computer where VALI software is installed. No further software is needed.

On top of the requirements described above, the customer must enable the connection at the network level: generally, customer network security policies do not interfere with the connection between ValiLink and MS-SQL.

ValiLink and all Belsim tools support Windows Authentication to connect to SQL server, in addition to SQL classic authentication.

6.5 ORACLE

ORACLE REQUIREMENTS SERVER SIDE

ValiLink is fully compliant with Oracle Database 8.0 or higher.

The Oracle server must be configured to enable reading and receiving data from VALI. A dedicated user with the appropriate rights must be created to enable the access to the Oracle database.

For further technical details regarding the Oracle server configuration, please visit Oracle support site at www.oracle.com.

ORACLE REQUIREMENTS CLIENT SIDE

ValiLink requires the following software installed on client computer for Oracle connection:

- Oracle9i Client Release 2 (9.2) or higher installed on client machine

The Oracle 9.2 Client Release must be installed on the computer where VALI software is installed. No further software is needed.

On top of the requirements described above, the customer must enable the connection at the network level: generally, customer network security policies do not interfere with the connection between ValiLink and the Oracle server.

TNSNAMES.ORA FILE CONFIGURATION

First, you need to configure the file tnsnames.ora to allow ValiLink and Oracle client software to connect to the database server. There are two ways to configure the file: by using the "Net Manager"

tool given with the Oracle Client installation or by editing manually the file tnsnames.ora

Using the "Net Manager" tool given with the Oracle Client installation.

- start Oracle NetManager:

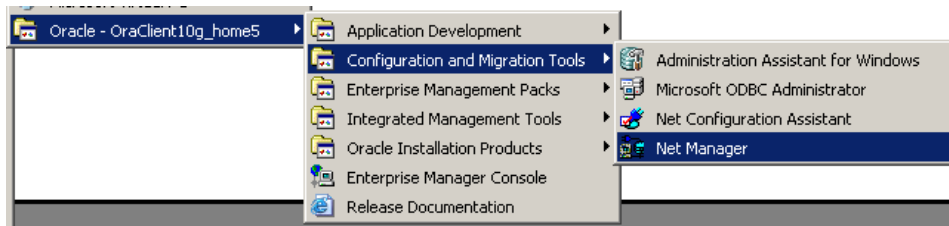


FIGURE 10 - STARTING ORACLE NET MANAGER

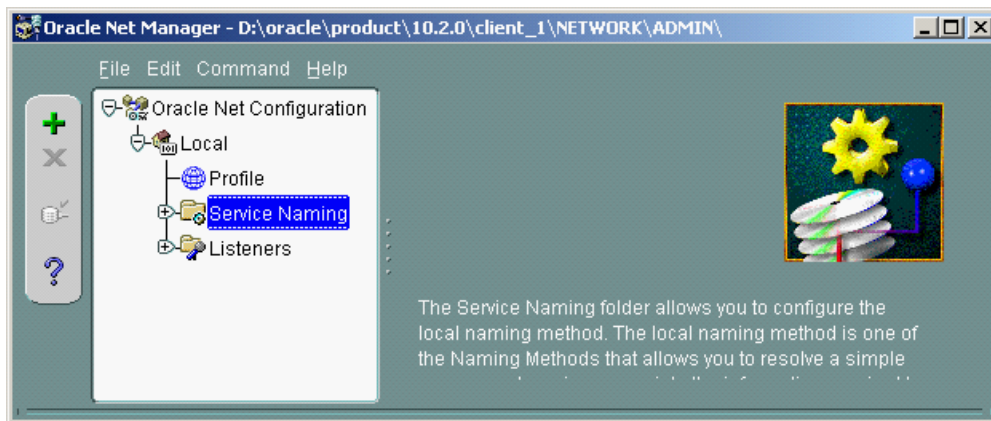


FIGURE 11 - ORACLE NET MANAGER

- click on the left arrow to create a new alias in the tnsnames.ora file:

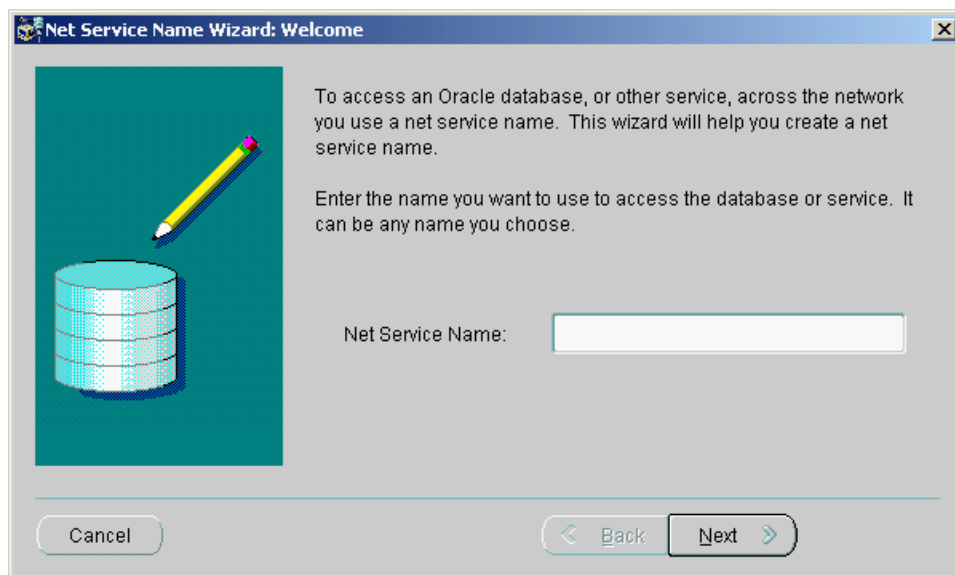


FIGURE 12 - DEFINE NAME OF NETWORK SERVICE

- enter the name you want to use for your alias:

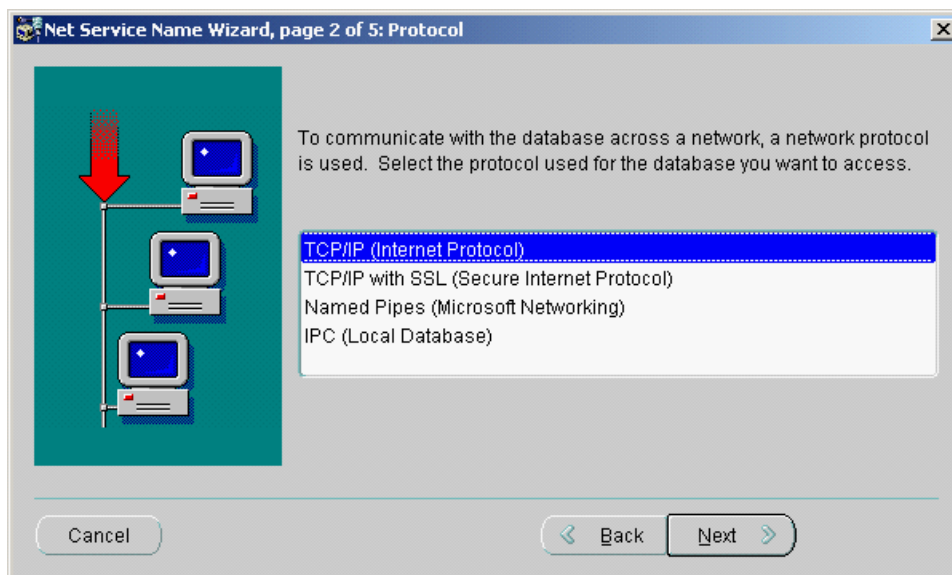


FIGURE 13 - TYPE OF COMMUNICATION PROTOCOL

- choose the type of communication protocol (TCP/IP by default):

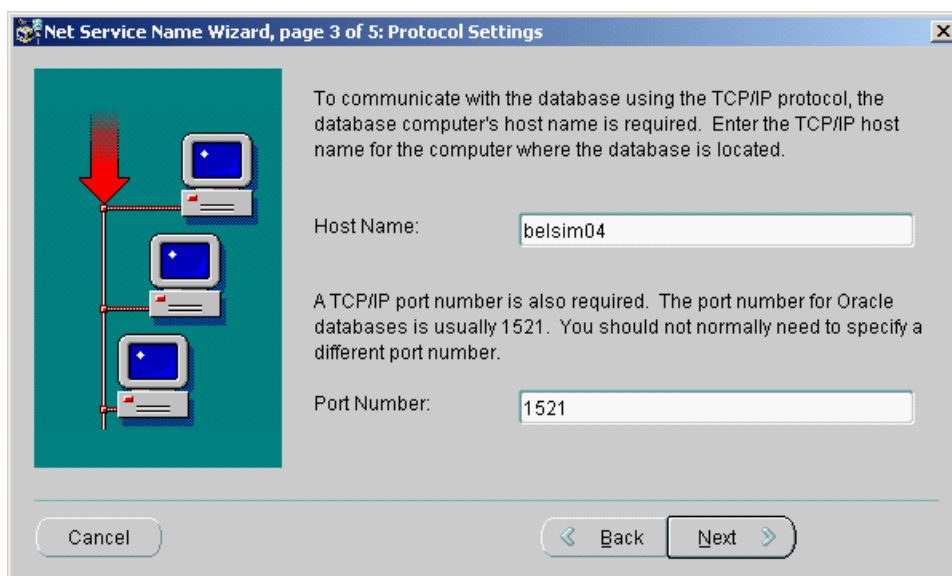


FIGURE 14 - IP ADDRESS OF THE ORACLE SERVER

- enter the IP address or UNC name of the Oracle Database Server:

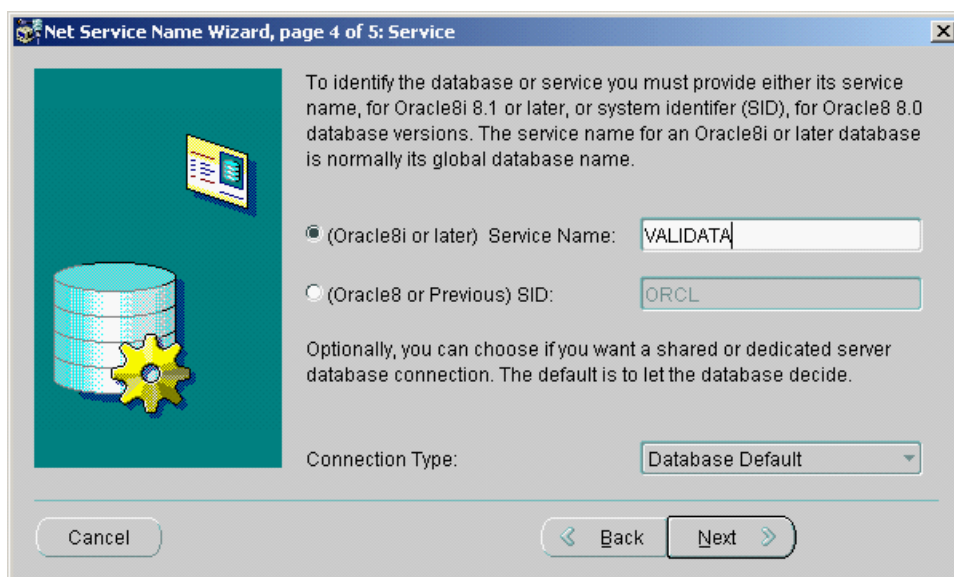


FIGURE 15 - NAME OF ORACLE DATABASE TO CONNECT TO

- enter the name of your active database or choose the SID:

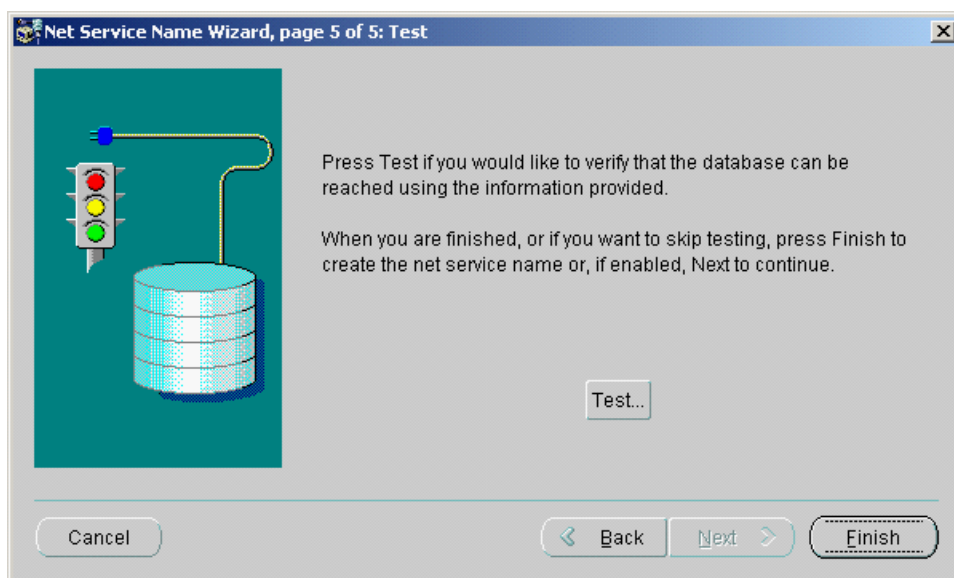


FIGURE 16 - TEST CONNECTIVITY

- finish the configuration by testing the connectivity with the Oracle Database Server

Editing manually the file tnsnames.ora

You can manually edit the file tnsnames.ora to add your own alias. The file is located in the directory Oracle installed on the client machine:

[InstallationDir]\NETWORK\ADMIN\TNSNAMES.ORA

Example of alias in the TNSNAMES.ORA file:

```
VALIDATA =  
  (DESCRIPTION =  
    (ADDRESS_LIST =  
      (ADDRESS = (PROTOCOL = TCP) (HOST = belsim04) (PORT = 1521))  
    )  
    (CONNECT_DATA =  
      (SID = validata)  
    )  
  )  
)
```

The alias name will be used by ValiLink to connect to the database.

CONFIGURATION IN BELSIM LICENSESERVER

To connect to the Oracle database, ValiLink uses a "Plant" that must be created through Belsim LicenseManager application

Valilink allows using two types of connections to Oracle Server:

- Oracle .NET Data Provider: optimized for .NET application (as ValiLink)
- ODBC (Open Database Connectivity)
- enter the name of the new plant. (Ex: VALI_PHD)
- in Server Type, choose "ODBC_Oracle" or "Oracle"
- enter the log name and password as defined in Oracle
- in IP/UNC, enter the name of the alias as defined in TNSNAMES.ORA

6.6 PHD

PHD REQUIREMENTS SERVER SIDE

The ValiLink plug to PHD is based on the standard OPC-HDA. The OPC server of PHD must be installed and configured to allow the connection using ValiLink. The PHD administrator must configure the security settings in PHD to allow connection and read-write access.

For further technical details regarding the PHD server configuration, please contact HoneyWell technical support.

PHD REQUIREMENTS CLIENT SIDE

If ValiLink is deployed on a computer without an installed OPC server then it may be necessary to install the OPC Core Components.

Install the core components setup file OPC Core Components 2.00 Redistributable 2.00.msi (available in [Belsim InstallDir]\SystemNT) on the computer where ValiLink runs and restart the computer.

On top of the requirements described above, the customer must enable the connection at the network level: generally, customer network security policies do not interfere with the connection between

ValiLink and the PHD OPC server.

CONNECTION TO OPC PHD SERVER

You must create a new plant in Belsim LicenseServer through Belsim LicenseManager.

- enter the name of the new plant. (Ex: VALI_PHD)
- in Server Type, choose "OPC_HDA_PHD"
- leave log name and password blank
- in IP/UNC, enter the IP address or name of the computer where the OPC server is installed

Enter the instance name of the server as parameter Database of the Server settings section of the ini file.

Example:

Database=OPC.PHDServerHDA.1

6.7 MATRIKON OPC HDA SERVER

OPC HDA REQUIREMENTS SERVER SIDE

The OPC HDA server of Matrikon must be installed and configured to allow the connection using ValiLink. The OPC HDA administrator must configure the security settings to allow connection and read-write access.

For further technical details regarding the OPC HDA server configuration, please contact Matrikon technical support.

OPC HDA REQUIREMENTS CLIENT SIDE

If ValiLink is deployed on a computer without an installed OPC server then it may be necessary to install the OPC Core Components.

Install the core components setup file *OPC Core Components 2.00 Redistributable 2.00.msi* (available in [Belsim InstallDir]\SystemNT) on the computer where ValiLink runs and restart the computer.

On top of the requirements described above, the customer must enable the connection at the network level: generally, customer network security policies do not interfere with the connection between ValiLink and the OPC HDAserver.

CONNECTION TO OPC HDA SERVER

You must create a new plant in Belsim LicenseServer through Belsim LicenseManager.

- enter the name of the new plant. (Ex: VALI_OPC)
- in Server Type, choose "OPC_HDA_MATRIKON"
- leave log name and password blank
- in IP/UNC, enter the IP address or name of the computer where the OPC server is installed

Enter the instance name of the server as parameter Database of the Server settings section of the ini file.

Example:

Database=OPC.ServerHDA.1



belsim

CHAPTER 14

VALIFILTER AND VALIEVENT

SECTION 1: INTRODUCTION

The main purpose of ValiFilter is an intelligent filtering of raw data to give an adequate set of input measurement data to VALI so that convergence rate is maximized.

ValiFilter analyses and filters raw measurement data before the execution of the validation step of ValiEngine or ValiOnlineEngine. It provides the following functionality:

- filtering measured values falling outside a given range
- setting OFF units or streams when a measured value falls under a given CUTOFF value
- defining specific filtering strategies using Flex code

ValiEvent aims at:

- the detection of process operation mode and logging of changes
- the detection and reporting on events on a given period, an event being a scatter, constant gradient (slope) or discontinuities (sudden changes) in the trend of the corresponding measured values

SECTION 2: VALIFILTER

ValiFilter is integrated with ValiEngine. It filters measured values, ensuring that they are within the alarm bounds defined on the Tags. It also enables to cut-off streams or units when the measured value is below a given cut-off values.

The syntax FILTERED enables to manually filter any measurements (ValiFilter license required) from a measurement file.

Eventually, specific filtering strategies can be defined through the use of Flex code.

2.1 FILTERING TAGS

When a measured value is outside the alarm bounds (LL, HH) the measured value, the precision and the failure probability are replaced as indicated by the user. The calculation of VALI will be based according to these replacement values.

A tag will be considered as filtered if a filter, a control string or a Flex Filter code (SETAGVAL function) has modified its value or has set it's accuracy to OFF.

You can also filter tags manually, replacing the measured value and optionally the accuracy by some manual input using the syntax FILTERED in measurement files (see "[Using measurement files](#)").

2.2 DEFINING THE FILTERS USING VALIMODELLER

The filter settings are defined in the Filter page of the Tags notebooks.

The **LL** and **HH** fields define the acceptable range of values for the measured value.

The **Value** field defines the action to be performed on the measured value if it is outside its acceptable range. It can be one of the following keywords:

- **NONE**: no action is taken
- **BND**: the measured value is replaced by the activated bound
- **DEF**: the measured value is replaced by the default value
- **MID**: the measured value is replaced by the average of the bounds $(LL+HH)/2$
- **LAST**: the measured value is replaced by the validated value of the corresponding tag (as obtained during the last successful and saved run)
- Value Physun: the measured value is replaced by the given value

The **Accuracy** field defines the action to be performed on the measured accuracy if the measured value is outside its acceptable range. It can be one of the following keywords:

- **NONE**: no action is taken
- **OFF**: the tag is set OFF
- **DEF**: the accuracy is replaced by the default value
- **MID**: the accuracy is replaced by half the difference between the bounds $(HH-LL)/2$
- **Value**: the accuracy is replaced by the given value (a negative value corresponds to a relative accuracy expressed in %)

The **Type** defines the type of the filtered accuracy when it is specified as a value.

The **FailProb** field defines the action to be performed on the failure probability if the measured value is outside its acceptable range. It can be a value or **NONE**, in which case no action is taken on the failure probability.

The **CUTOFF** field defines the threshold value for the measured value under which the corresponding stream or unit will be set OFF. It can be a value or **NONE**, in which case no CUTOFF action is taken.

Tags specified in project MAIN

Measurements | Definition | Bounds | **Filters** | Compens | KPI

Mask: **Filters** | Number of tags displayed: 2/18

Name: | Type: | PFD: | Stream/Unit: |

Tagname	Phys. Unit	LL	HH	Value	Accuracy	Type	Fail. Prob.	Cut Off	F
▶ AIR.1_T	C	-20	40	BND	10	A	NONE	NONE	☒
POWER_POWER	kw	10000	17000	DEF	5	%	50	NONE	☒

Stream: AIR.1 | T | Plant: VALI_PLANT

Physical Units: C | Measured: 15 | Accuracy: 1 | Type: A | Failure Probability: 5 %

Defaults: C | 15 | 1 | A | 5 %

Bounds: SL -273.15 | LL -20 | L -20 | H 40 | HH 40 | SH 5000.1

Comment:

New Copy Move Edit Rename Delete Details <<

OK Cancel Help

FIGURE 1 - THE FILTERS OF THE TAGS NOTEBOOK

2.3 DEFINING THE FILTERS USING MEA FILES

Attention: ValiFilter is available under a separate licence agreement.

ValiFilter detects measured values that are outside ERROR bounds (LL-HH interval) and updates their value, precision and failure probability according to user settings. These settings are defined in the model file. They can be updated using one of the two following syntaxes.

TAGNAME **FILTER** MEASURED ACCURACY FAILPROB

where:

- TAGNAME: the Tag name
- MEASURED: the action on the measured value and can be:
 - **NONE**: no action is taken
 - **BND**: the measured value is replaced by the activated bound
 - **DEF**: the measured value is replaced by the default value
 - **MID**: the measured value is replaced by the average of the bounds (LL+HH)/2
 - **LAST**: the measured value is replaced by the validated value of the corresponding tag (as obtained during the last successful and saved run)
 - Value Physun: the measured value is replaced by the given value
- ACCURACY: the action on the accuracy and can be:
 - **NONE**: no action is taken
 - **OFF**: the tag is set OFF
 - **DEF**: the accuracy is replaced by the default value
 - **MID**: the accuracy is replaced by half the difference between the bounds (HH-LL)/2
 - Value: the accuracy is replaced by the given value (a negative value corresponds to a relative accuracy expressed in %)
- FAILPROB: defines the action on the Failure probability and can be:
 - **NONE**: no action is taken
 - Value %: the failure probability is replaced by the given value

Example:

```
POWER_POWER LL 10 HH 17 MW
POWER_POWER FILTER DEF -5 50 %
AIR.1_T LL -20 HH 40 C
AIR.1_T FILTER BND 10 NONE
```

The second syntax allows to update several tags in one shot by using wildcards:

UPDATETAG	TYPE	TAGTYPE	FILTER	MEASURED	ACCURACY	FAILPROB
UPDATETAG	NAME	TAGNAME	FILTER	MEASURED	ACCURACY	FAILPROB

where:

- TAGTYPE and TAGNAME are strings defining which tags are to be changed (based on their type or name) and can contain wildcards

Example:

```
! Changes
01LCQ01CF001 FILTER 0.1 t/h -5 NONE
! Sets the filter settings of all tags starting by TI to MID MID NONE
!
UPDATETAG      NAME      TI*      FILTER MID MID NONE
```

Attention:

1. The penalty reported in the model and in the result files will be calculated both on basis of the raw measured value as on basis of the filtered value.
2. The control strings and test on CUTOFF value are based on the raw measured value

It is also possible to specify a **CUTOFF** value on the filter settings of an individual tag (ValiFilter license required). If the value of the measurement is below the specified value, the corresponding object (STREAM or UNIT) will be set OFF. The syntax is:

TAGNAME	CUTOFF	VALUE	PU
TAGNAME	CUTOFF	NONE	

where:

- TAGNAME: is the TAG name
- VALUE: is the value under which the stream will be set OFF
- PU: is the physical unit

Attention: It is also possible to specify a CUTOFF value on an individual tag via control strings (see ["Using control strings"](#)).

Remember also that the STATUS measurement enables a binary tag value to set a stream OFF.

2.4 INTERACTION WITH CUTOFF AND CONTROL STRINGS

The CUTOFF operation whether defined as a filter or through a control string uses the measured value (before filtering), except in case of manual filtering through a measurement file, in which case the filtered value is used.

More generally, if a control string is defined on the tag, its operation will be based on the measured value and it will over-write whatever has been made by the filter.

On the other side, if a control strings changes the value or the accuracy of a tag, the latter will be considered as filtered.

2.5 REPORTS

VALI flags with a F (in the reports r4v, r5v and r6v) the measurements on which automatic actions have been taken. The "measured" value is here the used value (as used by ValiEngine for the calculations).

Example:

```
*****
* TAGS OF THE SYSTEM *
*****
```

	TAG NAME	MEA.VAL.	MEA.ACC.	REC.VAL.	REC.ACC.	PENALTY	P.U.
	AIR.1_MFO2	21.000	CST	21.000			%
	AIR.1_MOLF	0.21500E+06	10.0	% 0.18372E+06	3.03	% 2.12	Nm3/h
	AIR.1_P	1.0130	2.00	% 1.0067	2.00	% 0.10	bar
*F	AIR.1_T	-20.000	10.0	0.42280	6.95	4.17	C
	AIR.2_P	20.000	2.00	% 20.068	1.40	% 0.03	bar
	AIR.2_T	500.00	2.00	499.71	2.00	0.02	C
	C-1_EFFIC	77.000	5.00	% 70.280	3.77	% 3.05	%
	FUME.1_T		OFF	1211.8	0.840	%	C
	FUME.2_MFO2	14.000	2.00	% 13.957	0.952	% 0.02	%
	FUME.2_P	1.0200	2.00	% 1.0194	1.99	% 0.00	bar
	FUME.2_T	600.00	2.00	599.89	1.99	0.00	C
	NG_MFC1		OFF	100.00	0.00		%
	NG_MOLF	6000.0	2.00	% 6047.8	1.91	% 0.16	Nm3/h
	NG_P	20.000	2.00	% 20.068	1.40	% 0.03	bar
	NG_T	15.000	1.00	15.001	1.00	0.00	C
F	POWER_POWER	16.100	5.00	% 15.785	4.54	% 0.15	MW
	R-1_DP	0.10000	0.100E-01	0.99993E-01	0.100E-01	0.00	bar
	T-1_EFFIC	80.000	5.00	% 81.266	1.36	% 0.10	%

FIGURE 2 - TAGS OF THE SYSTEM

A dedicated section in the r4v and r5v reports lists all the tags that have been filtered. The penalty is there calculated on basis of the original measured value.

Example:

```
*****
* FILTERED TAGS *
*****
```

	TAG NAME	MEA.VAL.	MEA.ACC.	FIL.VAL.	FIL.ACC.	PENALTY	P.U.
*F	AIR.1_T	-40.000	1.00	-20.000	10.0	(1634.00)	C
F	POWER_POWER	20.000	1.00	% 16.100	5.00	% (444.09)	MW

FIGURE 3 - FILTERED TAGS

2.6 ADVANCED FILTERING STRATEGIES

Using Flex code you can define further filtering actions :

- to update measured values, accuracies or bounds of tags
- to set OFF units and streams
- to deactivate user defined equations

Please refer to the [Flex Language](#) chapter of ValiModeller Manual for further information.

SECTION 3: VALIEVENT

3.1 OPERATION

The application (ValiEvent.exe) processes trends of measured values used in a VALI model. It aims at detecting events such as scatter, constant gradient (slope) or discontinuities (sudden changes) in the trend of the corresponding measured values in order:

- to detect process unsteady state (ValiPerformance application)
- to detect changes in the operation modes of some production units (ValiProduction application)

Any event in the trend of one of the selected measurements is reported in an html report which is linked to the main VALI reports

Three different types of events are detected:

- variance (or scatter): this event is detected on basis of an acceptable scatter of the followed variable around its average value. The comparison is made on an individual standard deviation of the tag, the estimation being performed on the sample of measured data of the followed variable. An event is detected when this standard deviation (sigma) is larger than the threshold value

Attention: The standard deviation (used as threshold) is expressed in absolute (not relative) and is not necessarily the value used by VALI (which is not known by ValiEvent).

- gradient (or slope): this event is detected on basis of an acceptable slope of the linearised trend () of the followed variable. An event is detected when the absolute value of the slope $|b|$ is larger than the threshold value
- discontinuity (or sudden change): this event is detected by a statistical analysis of the second derivative of the signal. If the biggest and/or lowest values of the latter function are away from the average value by more than a configurable number (Criter) of times the standard deviation, an event is detected. A configurable number of highest (NvalHi) and lowest (NvalLo) values are eliminated before calculating the average and the standard deviation

Attention: If all the measured values are within 1 sigma (threshold value) around the average value of the sample, then the third criterion (discontinuity) is not computed.

For each event detected, an excel readable file (csv file) is created. Opening it with excel enables to easily create charts of the values of the measurement.

3.2 STARTING VALIEVENT

ValiEvent is usually launched by ValiScheduler. It can also be started manually or by an external application by starting the application ValiEvent in <installdir>\bin directory (by default C:\Program Files\Belsim\bin).

It requires as argument the name of the INI file with the configuration settings as well as the list of tags with their acceptable standard deviation and gradient.

Here is the format of the **TagList** input file (that can be used also as input file of ValiLink):

```
TAGNAME      STDEV      [ GRADIENT ]
```

where:

- TAGNAME: the TAG name
- STDEV: the acceptable scatter expressed as a standard deviation (absolute value)
- GRADIENT: the acceptable gradient (or slope) expressed as a delta over the dimension of the time interval

The 2 first columns (TagName and Sigma) are mandatory but the 3rd column (Gradient) is optional: if there is no Gradient, a default value is taken as: $GRADIENT = (2 * STDEV)$ over the dimension of the time interval.

The report (html format) is made of three sections:

- tags not found: tags for which no value can be retrieved from the external system
- interpolations on tags: tags for which not all values can be retrieved from the external system, missing data are replaced by interpolated values.
- events detected: tags for which an event has been detected. The criteria that is not satisfied is pinpointed in red

Example of ValiEvent report:





RUN INFORMATION

Summary

License	BELSIM S.A. #99
Date	13/04/2012
Time	16:10:34
User	hugues
Version	4.7.0.3
Ini file	ValiEvent.ini
Data file	VALIMODELLER\R-SQL-EVENTS2.MEA
Tags file	TAGSF0REVENTS.TXT

Parameters

Duration	4.00
Duration unit	H
Maximum number of values	240

Results

Number of expected tags	4
Number of analyzed tags	4
Number of missing tags	0
Number of bad tags	0
Number of interpolated tags	0
Number of tags with events	2

EVENTS DETECTION

	Graph	Tag Name	Accept. StDev	StDev	Accept. Gradient	Gradient	Accept. Disc.	Disc.	Data for Excel
*		MolarFlow1	5000.0	6800.6	500.00	202.77	2.0000	3.5397	MolarFlow1.csv
		MolarFlow2	500.00	198.72	250.00	14.199	2.0000	2.7704	MolarFlow2.csv
		MassFlow1	250.00	99.360	125.00	7.0993	2.0000	2.7704	MassFlow1.csv
*		Temperature1	10.000	62.187	5.0000	9.5681	2.0000	6.3724	Temperature1.csv

MOLARFLOW1

3.3 CONFIGURING VALIEVENT

The event detector is configured through the file ValiEvent.ini which is formatted according to Windows INI files and has two sections **[INIT]** and **[DIR]**. It can also be configured directly from ValiScheduler.

The section **[INIT]** specifies the settings of the statistical analysis:

- **CRITER:** value of the criteria for detecting an event
- **NVALHI:** number of higher values to be ignored
- **NVALLO:** number of lower values to be ignored
- **NVALUE:** maximum number of values that will be analyzed for each tag
- **DURATION:** the time interval on which the data are taken
- **DIMENSION:** the physical units of the duration: S, s, SEC, Sec, sec, M, m, MIN, Min ,min, Hour, H, h, HOUR, Hour, hour, Day, D, d, DAY ,Day, day

Attention: Default values for duration and dimension are respectively 1 day

The section [DIR] specifies the name of the input files as well as the working folder for the CSV files:

- **DATAFILE:** name of the input file with the measurement data
- **CSVDIR:** directory where CSV file will be created
- **TAGLIST:** name of the input file with the list of tags to be analyzed including their acceptable standard deviation and gradient

Example:

```
[Init]
Criter= 6.0
NvalHi= 4
NvalLo= 4
Nvalue= 100
Duration= 2
Dimension= h

[DIR]
DataFile=events.mea
CsvDir="C:\Vali\PlantA\events"
TagList=input.txt
```

Attention: If ValiEvent is launched from ValiScheduler, entries marked with a (*) have to be introduced manually in the initialization file (ValiEvent.ini).



belsim

CHAPTER 15

VALI ONLINE ENGINE

SECTION 1: INTRODUCTION

ValiOnlineEngine is the online version of the computation tool of VALI. It is optimized to run in an automated procedure generating online validation results.

There is no difference between the ValiOnlineEngine and the ValiOfflineEngine integrated in ValiModeller with regard to the performance of the tool or the results of the calculation. The main difference is that the operation of the ValiOnlineEngine can be automated either through its associated ValiScheduler or through any external automation procedure.

ValiScheduler controls the sequence of the different VALI components:

- getting the measurement data from a DCS, a PIMS or any other databases (ValiLink)
- filtering the measurement data (ValiFilter, which is integrated into the ValiOnlineEngine)
- analyzing the measurement data in order to detect events (ValiEvent)
- running ValiOnlineEngine
- writing back the results to a PIMS or any other database (ValiLink)
- exporting validation results to ValiDataBase (ValiReport)
- backing-up the files generated during the run

Although ValiScheduler is devoted to automated operation it also features a manual mode that can be used for testing purpose.

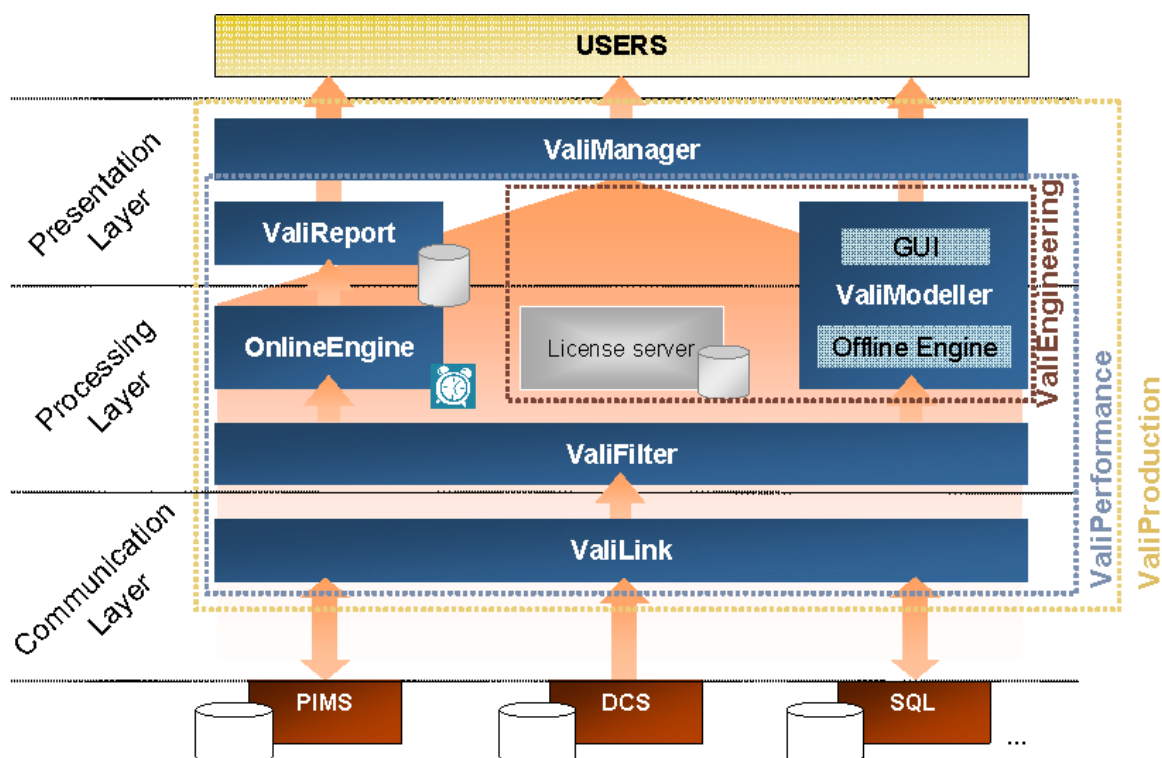


FIGURE 1 - VALIONLINEENGINE ENVIRONMENT

SECTION 2: RUNNING VALIONLINEENGINE

2.1 STARTING VALIONLINEENGINE

ValiOnlineEngine has been designed to be automated through its coupled ValiScheduler. However it can also be automated through other applications.

To start ValiOnlineEngine:

- start the application Vali.exe in the <installdir> folder (by default Program Files\Belsim\bin)
- double-click on a pc.vif file containing the settings of the run

2.2 VALI INFORMATION FILE (pc.vif)

When starting, ValiOnlineEngine needs to know which model has to be run as well as the TAG and MEA files to be used.

This information is defined in a Vali Information File (default extension vif). If not given as argument, VALI will look by default for a file named **pc.vif**. If this file is not present in the current folder, the information will be requested interactively.

Contents of the VALI information file:

- line 1: name of the model file
- line 2: name of the model (PFD)
- line 3: runtime options
- line 4: name of the Flex file
- line 5: name of the MEA file
- line 6: name of the TAG file
- line 7: name of the model back-up file
- line 8: name of second MEA file
- line 9: name of third MEA file

Example:

```
gt.bls
GT
-htm -sqp
```

```
GT.MEA
GT.TAG
```

For more information about the runtime options, see "[How to run your data validation model](#)".

2.3 RUN STATUS

At the end of a run, VALI produces a file named **valiexit.log** with one line containing the keyword EXITCODE followed by a numeric code:

- -3: system error - no connection to the license server or licence issue
- -2: system error - installation issue: missing directory, wrong version number
- -1: fatal error
- 0: the run was successful
- 1: the run was successful but gross errors are suspected to be present (according to Chi-square test)

Example:

```
EXITCODE 0
```

SECTION 3: RUNNING VALISCHEDULER

3.1 STARTING VALISCHEDULER

To start the scheduler, you can either:

- start the ValiScheduler via the Windows Start Menu: (**Start Menu | Programs | Belsim | Vali4 | ValiScheduler**)
- double-click the executable "ValiScheduler.exe" in the **bin** sub-folder of Belsim installation folder

At start up, the application will ask whether to connect to a remote ValiService or to open a local ValiScheduler configuration file.

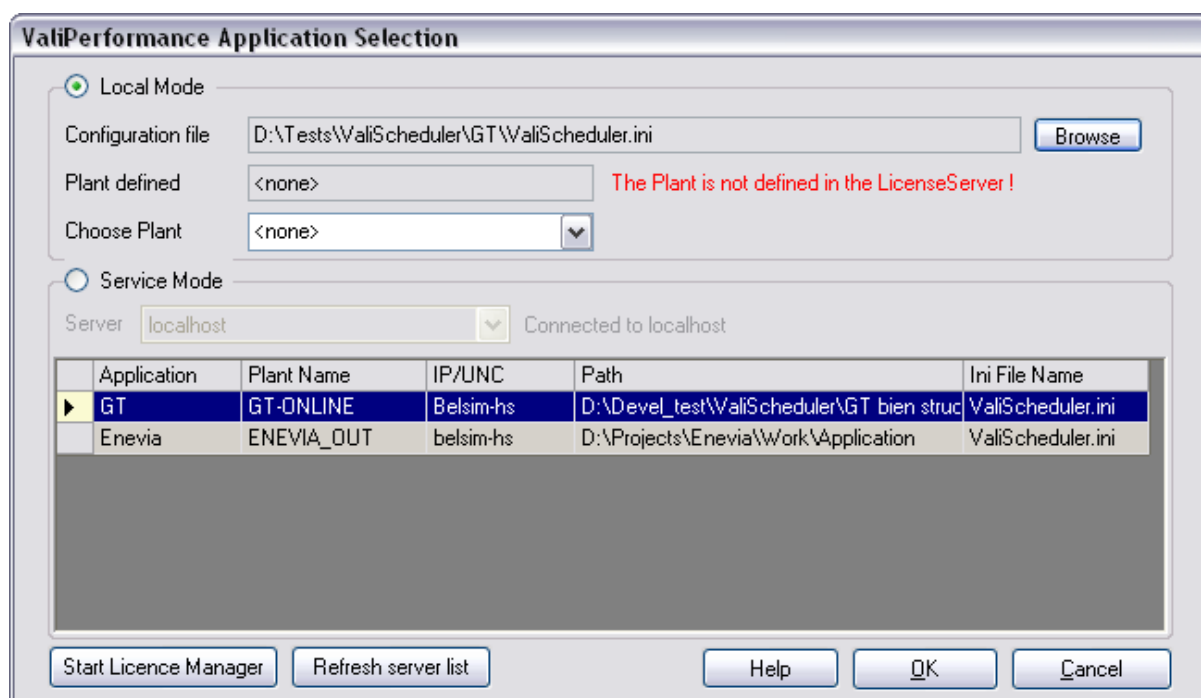


FIGURE 2 - START-UP PAGE OF VALISCHEDULER (INI FILE)

Using local mode, an INI file will be requested. To create a new application, you simply need to enter a new name for the INI file. To open an existing file, simply browse to open it.. ValiScheduler also requires the name of a Plant defining a connection to a ValiDatabase. This allows ValiScheduler to create all runs in the database when starting to process runs.

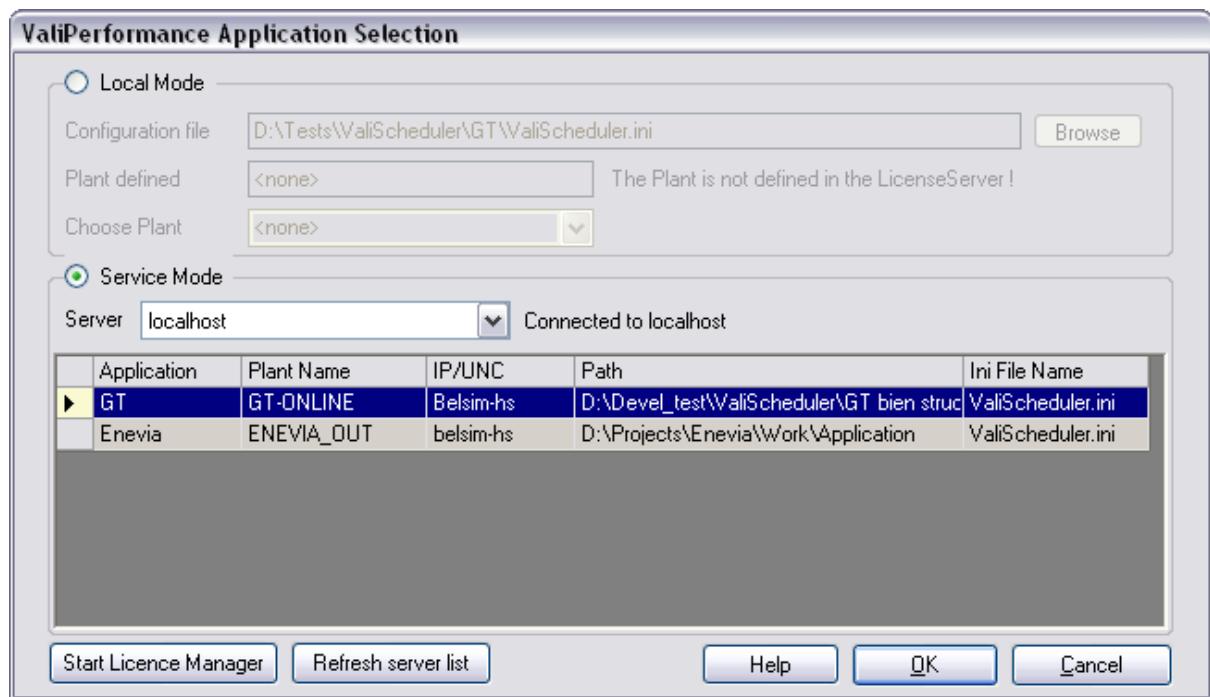


FIGURE 3 - START-UP PAGE OF VALISCHEDULER (VALISERVICE)

Using service mode, ValiScheduler will open a TCP/IP connection with the ValiService application and retrieve the settings remotely. The IP address of the server where application is defined can be selected using the drop-down list (see "[Belsim LicenseManager](#)" to add new IP addresses).

When using ValiScheduler coupled to ValiService, the tasks will be processed by the computer where ValiService is installed.

The settings of ValiScheduler are accessed via a notebook that contains 8 pages:

- General: settings for scheduling either automated run or manual runs,
- Acquisition: options for getting measurement data from a DCS, a PIMS, or other databases,
- Reconciliation: ValiOnlineEngine settings,
- Storage: options for storing results of the validation process,
- Event: settings of the event detector,
- Backup: files or folders to be copied for back-up purpose,
- Log File: options for logging,
- Email: settings for automatic emailing in case of non convergence

3.2 GENERAL PAGE

The **General** page allows specifying the settings for scheduling either online run, re-run, or manual run as well as the steps to be performed:

- Real Time Run is the fully automated procedure to process real time data
- Re-Run is the automated procedure to reprocess runs from the past
- Manual Run is used to test the online/rerun procedure or parts of this procedure

Attention: Real Time runs and Reruns can be performed at the same time. Manual run can be started once Online runs and Reruns are stopped.

The "Overview" table is used to show to the user an overview of the whole process. The top of the page is divided in three parts:

- Real Time Runs
- Reruns
- Manual Runs

For each option, different information items are displayed:

- Status: status of the process:
 -  : process is stopped
 -  : in real time mode, it indicates that some reruns are being processed and that online runs will be processed further
 -  : process is running. For real-time, the scheduler is online
- From: date of the first run to be processed
- To: date of the last run to be processed
- Last Run: date and time of the last processed run
- Progression: progress bar indicating the number of runs already processed compared to the number of runs to be processed. For real time processing, indicates the number of reruns to be processed before starting online processing

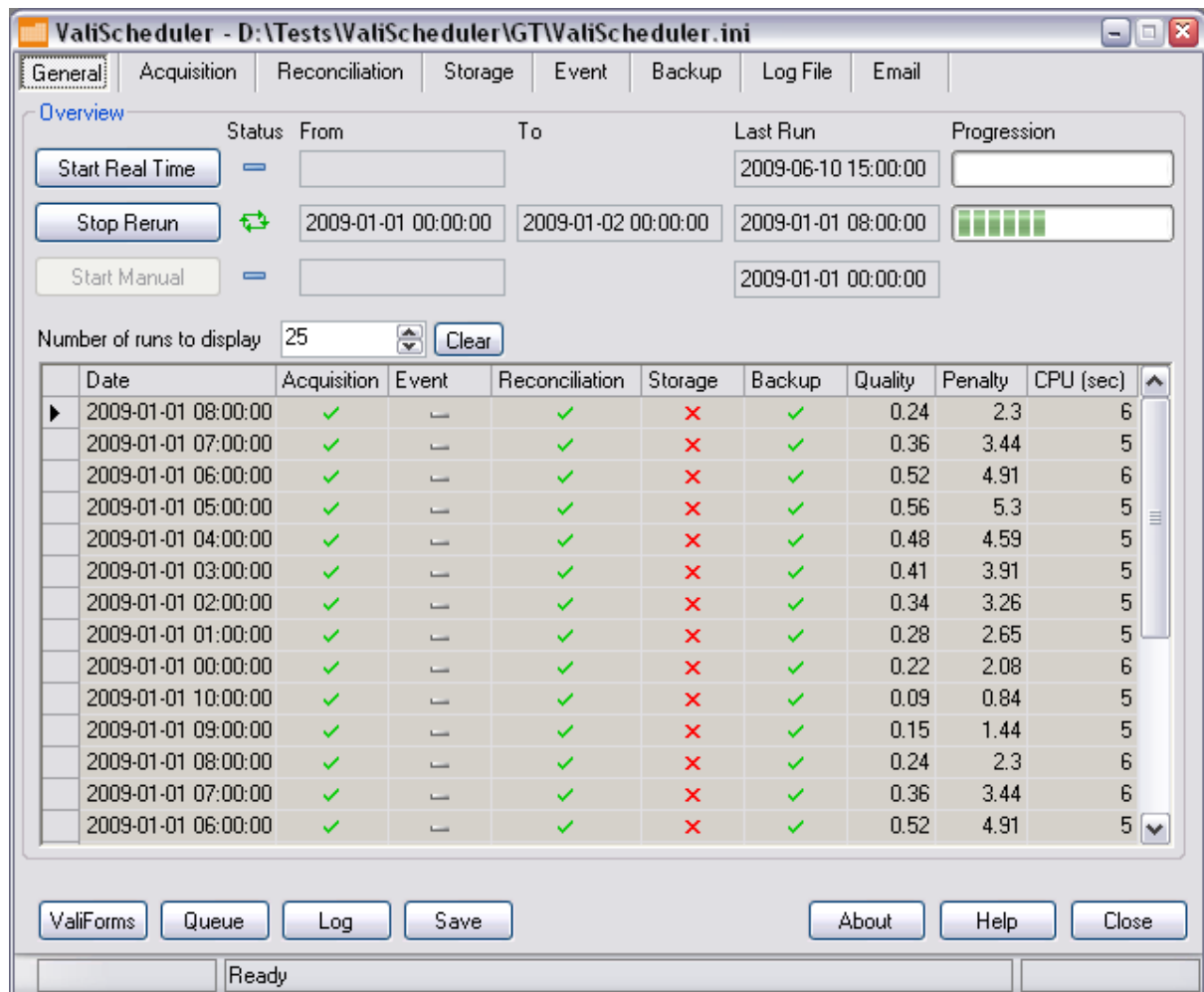
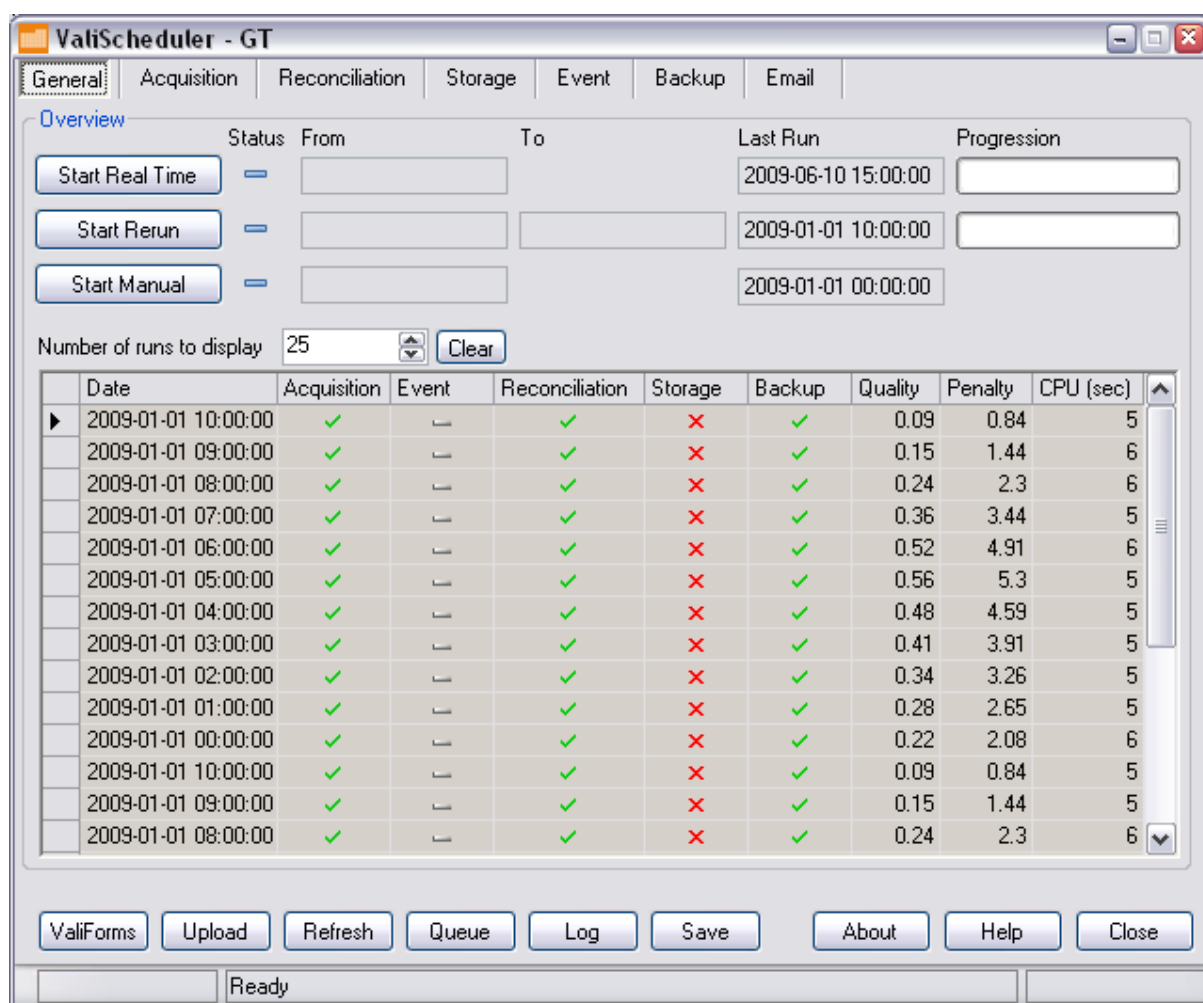


FIGURE 4 - GENERAL PAGE OF VALISCHEDULER USED LOCALLY

The table in the bottom of the page displays information about the last runs (Real Time/Rerun/Manual):

- Date: date of the run (based on the beginning of the time period)
- Acquisition/Event/Reconciliation/Storage/Backup :status of the step:
 -  : step not executed
 -  : step completed successfully
 -  : step failed
- Quality: quality of the run
- Penalty: penalty of the run
- CPU(sec): time (in seconds) taken to process the run



Date	Acquisition	Event	Reconciliation	Storage	Backup	Quality	Penalty	CPU (sec)
2009-01-01 10:00:00	✓	⬜	✓	✗	✓	0.09	0.84	5
2009-01-01 09:00:00	✓	⬜	✓	✗	✓	0.15	1.44	6
2009-01-01 08:00:00	✓	⬜	✓	✗	✓	0.24	2.3	6
2009-01-01 07:00:00	✓	⬜	✓	✗	✓	0.36	3.44	5
2009-01-01 06:00:00	✓	⬜	✓	✗	✓	0.52	4.91	6
2009-01-01 05:00:00	✓	⬜	✓	✗	✓	0.56	5.3	5
2009-01-01 04:00:00	✓	⬜	✓	✗	✓	0.48	4.59	5
2009-01-01 03:00:00	✓	⬜	✓	✗	✓	0.41	3.91	5
2009-01-01 02:00:00	✓	⬜	✓	✗	✓	0.34	3.26	5
2009-01-01 01:00:00	✓	⬜	✓	✗	✓	0.28	2.65	5
2009-01-01 00:00:00	✓	⬜	✓	✗	✓	0.22	2.08	6
2009-01-01 10:00:00	✓	⬜	✓	✗	✓	0.09	0.84	5
2009-01-01 09:00:00	✓	⬜	✓	✗	✓	0.15	1.44	5
2009-01-01 08:00:00	✓	⬜	✓	✗	✓	0.24	2.3	6

FIGURE 5 - GENERAL PAGE OF VALISCHEDULER USED WITH REMOTE CONNECTION TO VALISERVICE

Additional buttons are available at the bottom of the page:

- ValiForms: this button opens ValiForms. The xml file ("ValiForms.xml") describing the contents of the Forms has to be located next to the ValiScheduler INI file
- Upload: only available when working with ValiService, this functionality allows you to upload files located on your computer into the application directory located on the ValiService computer. This window can also be used to download files located in the application directory of the server

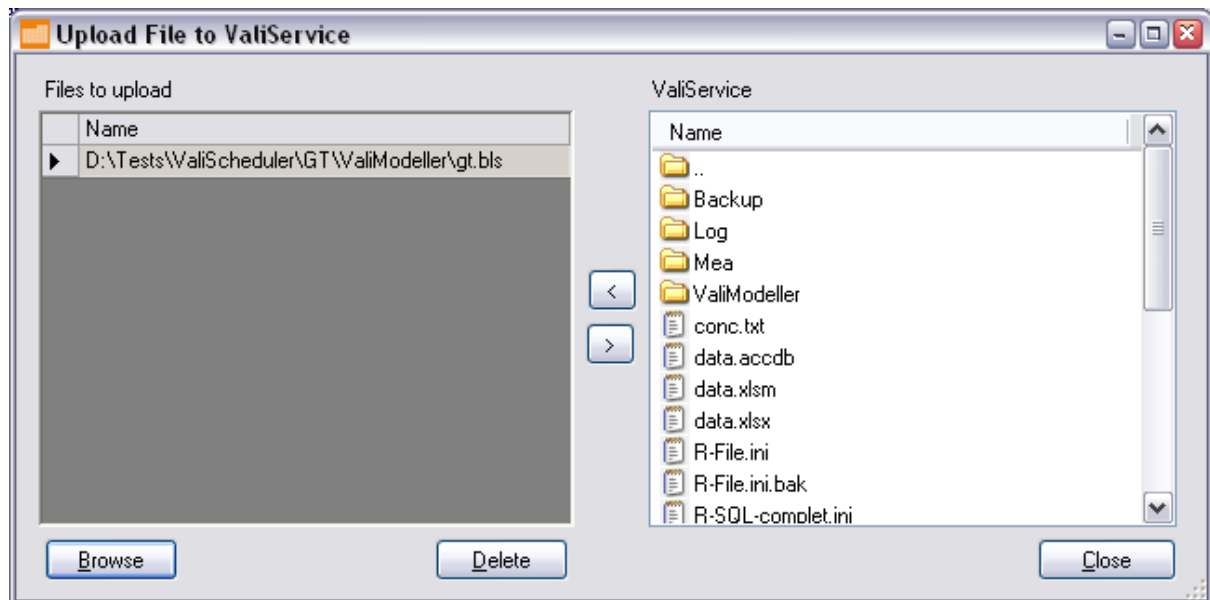


FIGURE 6 - UPLOAD PAGE OF VALISCHEDULER USED WITH REMOTE CONNECTION TO VALISERVICE

- Refresh: only available when working with ValiService, this functionality allows you to manually reload the content of the configuration file (run results, values...) and refresh the graphical user interface
- Queue: this functionality shows you an overview of the process queue list. This allows you:
 - to have a general overview of the process queue list
 - to see if other applications are running
 - to determine how many runs are waiting in the process queue list

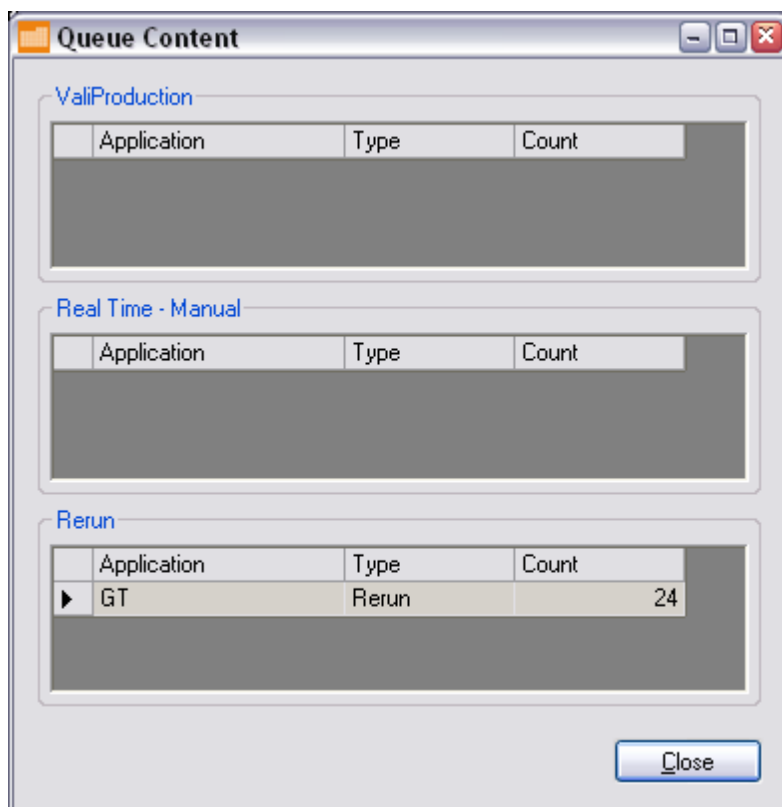


FIGURE 7 - VIEW QUEUE PAGE OF VALISCHEDULER USED WITH REMOTE CONNECTION TO VALISERVICE

- Log: When running ValiScheduler locally, the button opens a window that will display the content of the ValiScheduler log file. When ValiScheduler is used with remote connection to ValiService, the button opens a window that will display the content of the ValiService log file located on the remote server

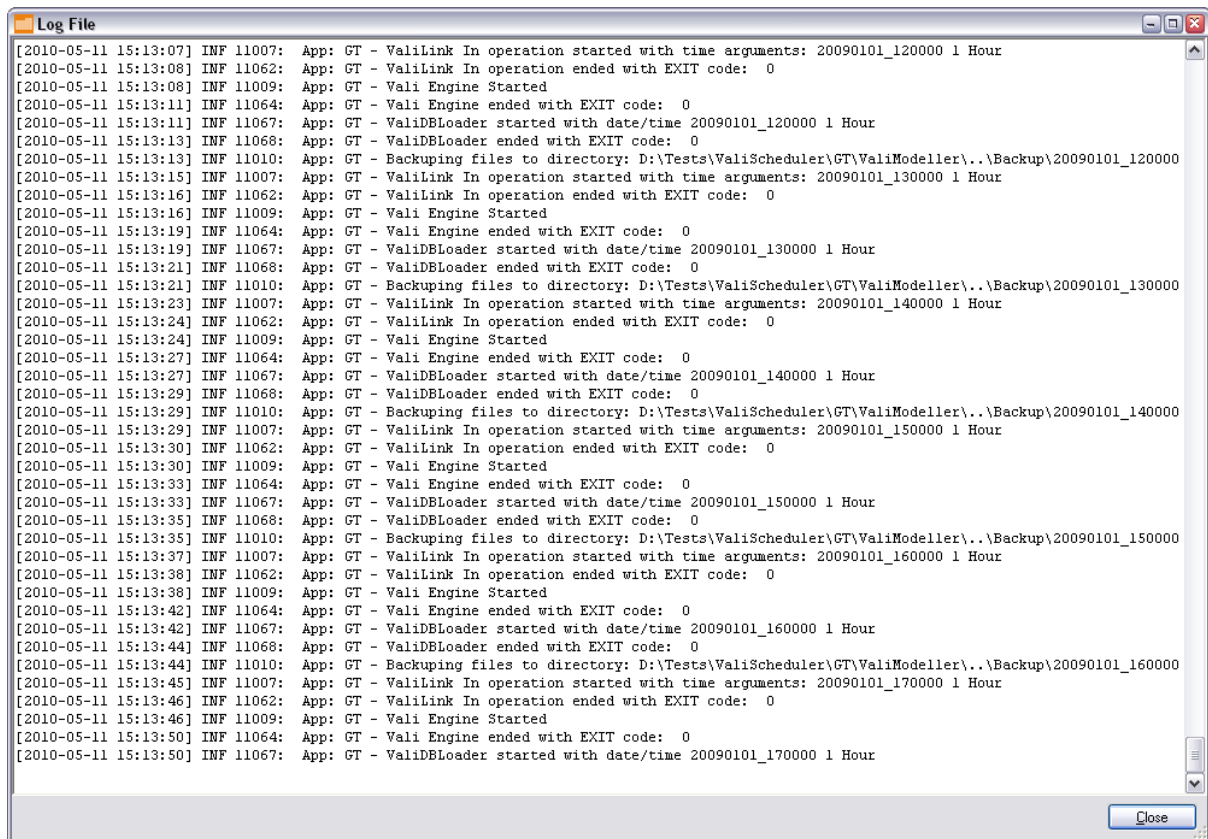


FIGURE 8 - VIEW LOG PAGE OF VALISCHEDULER USED WITH REMOTE CONNECTION TO VALISERVICE

The Validation results will be reloaded automatically every second. You can click on the button "Refresh" to refresh the results manually.

3.2.1 STARTING REAL TIME

To start the execution of real time tasks, click on "Start Real Time" button. A configuration window will be opened with the runs settings.

Start Real Time Operation

Start From

☒ Date: 2011-03-02 15:18:21

☐ Last Processed:

☐ Last Successful Run: 2011-03-02 00:00:00

☐ Now: 2011-03-02 15:00:00

☐ Rerun Only non converged Runs

Scheduling Settings

Run Every: 1 Hour

Agg. Period: 1 Hour

Time Delay: 0 Minute

TimeOut: -1 Minutes

Steps

☐ (Select All)

☐ Data Acquisition

☒ Online Mode

☐ Backup Mode

☐ Event Detection

☐ Data Reconciliation

☐ Data Storage

☐ Backup

☐ Email

OK Cancel

FIGURE 9 - START REAL TIME OPERATION PAGE OF VALISCHEDULER

The processing of the runs will start directly when you click on the "OK" button. The "Start Real Time" button of the general page will be renamed "Stop Real Time". Once official runs are under processing, you can not change the configuration settings. To stop the official run execution, click on the button "Stop Real Time". The button will be renamed "Stopping Real Time". If tasks are under execution, ValiScheduler will wait till the current tasks are finished.

The "Start From" table is used to specify the date of the first run to be processed:

- Date: when the first run will be started
- Last Processed: the date of the last processed run will be used for the first run
- Last Successful Run: the date of the last successful run will be used for the first run
- Now: ValiScheduler calculate the next real time date to process. This date will be used for the first run
- Rerun Only non Converged Runs: if selected, ValiScheduler will use the status of the run specified in the associated ValiDatabase (defined by the Plant) to determine if the run must be processed or not

If the start date specified is different from "Now", all the runs processed in the past will be considered as "Reruns". Once ValiScheduler has processed the old runs and is back online, the next processed run will be considered as "Real Time".

The "Schedule Settings" table is used to specify the settings of the automated procedure:

- Run Every: at which frequency successive runs will be scheduled
- Agg. period: the time frame for aggregating the data from historian system. For example, one can be interested to make only one run per day, but based on hourly data (not daily). In this case, the runs frequency should be set to 24 hour and the aggregation period to 1 hour. Another example is the case where one wants to validate every hour the data from the past 24 hours. In this case the runs frequency should be set to 1 hour and the aggregation period to 24 hours
- Time Delay: for real time runs, waits the time specified before starting execution of the tasks. This option is useful to ensure that all data are available in the PIMS before starting the acquisition
- Time Out: time period allowed for each task of the run. If the task lasts for a longer time period the task is cancelled. No time out will be applied if this value is negative. During the reconciliation step, ValiScheduler publishes a file next to the model file, called VALISTOP, once the timeout is reached. During resolution, ValiEngine will check for the existence of such a file. If it exists, ValiEngine will stop its resolution, publish the reports and close. If ValiEngine takes more than 2 minutes between the publishing of VALISTOP and its closing, ValiScheduler kills ValiEngine process and free its license (as well as the ValiFilter license)

The "Steps" table is used to specify the tasks that will be executed in the automated procedure:

- Data Acquisition: If selected, ValiScheduler will launch the retrieval of measurement data from DCS, PIMS, or other databases:
 - Online Mode: Start acquisition tasks defined in "Acquisition" page with "Online Mode" selected
 - Backup Mode: Typically used with "Rerun", start tasks defined in "Acquisition" page with "Backup Mode" selected and restore backup files (See "[backup](#)" page for more details about restoring backup files)
- Event Detection: if selected, ValiScheduler will launch the event detection (ValiEvent) on measurement data (See [ValiEvent](#) documentation)
- Data Reconciliation: if selected, ValiScheduler will launch ValiOnlineEngine
- Data Storage: if selected, ValiScheduler will launch the storing of results after the validation step
- Backup: if selected, ValiScheduler will back-up the selected files after the validation step
- Email: if selected, ValiScheduler will launch the automatic emailing according to the parameters specified in the "[Email](#)" page

The time window of the validation tasks is delimited by two dates: BeginDate and EndDate. The difference between both dates corresponds to the Time Interval:

$$\text{EndDate} = \text{BeginDate} + \text{Interval}[\text{IntervalDim}]$$

The Time Interval is equals to the time span value of "Run every".

ValiScheduler will start the tasks execution when the [BeginDate + TimeDelay] is lower than the current local time. If the [BeginDate + TimeDelay] value is higher, ValiScheduler will wait till the local time is higher.

After a run is terminated, the next BeginDate will be set as the old EndDate.

When executing ValiScheduler without ValiService (using a configuration file), ValiScheduler needs to be open to execute tasks automatically. If you close ValiScheduler, the automatic execution of tasks will be stopped.

When executing ValiScheduler with ValiService, you can close the ValiScheduler application safely without altering ValiService operation. ValiService will start automatically from the last processed run in case of computer or network connection to the LicenseServer failure. When coming back in the graphical interface of ValiScheduler, all the parameters and the status of the process are automatically refreshed according to ValiService operation.

3.2.2 STARTING RERUN

To start the execution of rerun tasks, click on "Start Rerun" button. A configuration window will be opened with the runs settings.

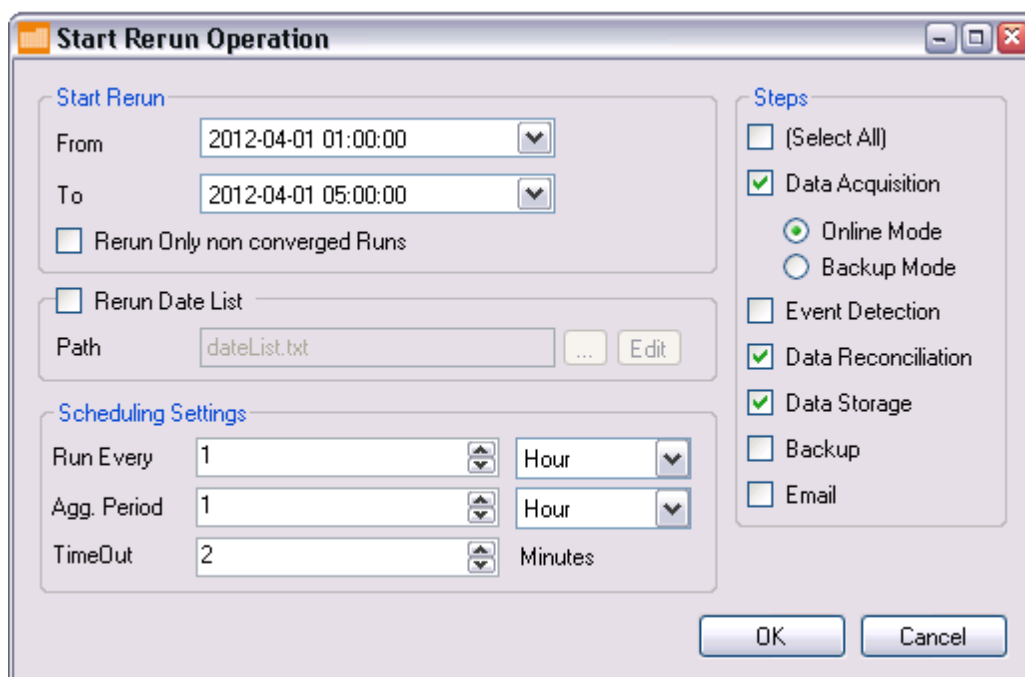


FIGURE 10 - START RERUN OPERATION PAGE OF VALISCHEDULER

The processing of the runs will start directly when you click on the "OK" button. The "Start Rerun" button of the general page will be renamed "Stop Rerun". Once runs are under processing, you cannot change the configuration settings. To stop the run execution, click on the button "Stop Rerun". The button will be renamed "Stopping Rerun". If tasks are under execution, ValiScheduler will wait till the current run is finished.

The "Start Rerun" table is used to specify the date of the first run:

- From: the date of the first run to be processed
- To: the date of the last run to be processed. After this run will be processed, ValiScheduler (or ValiService) will stop the processing of runs
- Rerun Only non Converged Runs: If selected, ValiScheduler will use the status of the run specified in the associated ValiDatabase (defined by the Plant) to determine if the run must be processed or not

The "Rerun Date List" table is used to specify in a text file a set of dates that have to be processed (mainly used for testing purpose). Dates are defined line by line in the input text file (one date per line). The date format must be YYYYMMDD_HHMMSS. If the "Rerun Date List" option is checked, then the classical dates selection (From / To) and the frequency of calculation (Run Every) are not used anymore. Other rerun options are still used.

The "Schedule Settings" table is used to specify the settings of the automated procedure:

- Run Every: at which frequency successive runs will be scheduled
- Agg. period: the time frame for aggregating the data from historian system. For example, one can be interested to make only one run per day, but based on hourly data (not daily). In this case, the runs frequency should be set to 24 hour and the aggregation period to 1 hour. Another example is the case where one wants to validate every hour the data from the past 24 hours. In this case the runs frequency should be set to 1 hour and the aggregation period to 24 hours
- Time Out: time period allowed for each task of the run. If the task lasts for a longer time period the task is cancelled. No time out will be applied if this value is negative. During the reconciliation step, ValiScheduler publishes a file next to the model file, called VALISTOP, once the timeout is reached. During resolution, ValiEngine will check for the existence of such a file. If it exists, ValiEngine will stop its resolution, publish the reports and close. If ValiEngine takes more than 2 minutes between the publishing of VALISTOP and its closing, ValiScheduler kills ValiEngine process and free its license (as well as the ValiFilter license)

The "Steps" table is used to specify the tasks that will be executed in the automated procedure:

- Data Acquisition: If selected, ValiScheduler will launch the retrieval of measurement data from DCS, PIMS, or other databases:
 - Online Mode: Start acquisition tasks defined in "Acquisition" page with "Online Mode" selected
 - Backup Mode: Typically used with "Rerun", start tasks defined in "Acquisition" page with "Backup Mode" selected and restore backup files (See "[backup](#)" page for more details about restoring backup files)
- Event Detection: if selected, ValiScheduler will launch the event detection (ValiEvent) on measurement data (See [ValiEvent](#) documentation)
- Data Reconciliation: if selected, ValiScheduler will launch ValiOnlineEngine
- Data Storage: if selected, ValiScheduler will launch the storing of results after the validation step
- Backup: if selected, ValiScheduler will back-up the selected files after the validation step
- Email: if selected, ValiScheduler will launch the automatic emailing according to the parameters specified in the "[Email](#)" page

The time window of the validation tasks is delimited by two dates: BeginDate and EndDate. The difference between both dates corresponds to the Time Interval:

$$\text{EndDate} = \text{BeginDate} + \text{Interval}[\text{IntervalDim}]$$

The Time Interval is equals to the time span value of "Run every".

After a run is processed, the next BeginDate will be set as the old EndDate.

When executing ValiScheduler without ValiService (using a configuration file), ValiScheduler needs to be open to execute tasks automatically. If you close ValiScheduler, the automatic processing of runs will be stopped.

When executing ValiScheduler with ValiService, you can close the ValiScheduler application safely without altering the ValiService operation. ValiService will start automatically from the last processed run in case of computer or network connection to the LicenseServer failure. When coming back in the graphical interface of ValiScheduler, all the parameters and the status of the process are automatically refreshed according to ValiService operation.

3.2.3 STARTING MANUAL RUN

To start the execution of manual runs, click on "Start Manual" button. A configuration window will be opened with the runs settings.

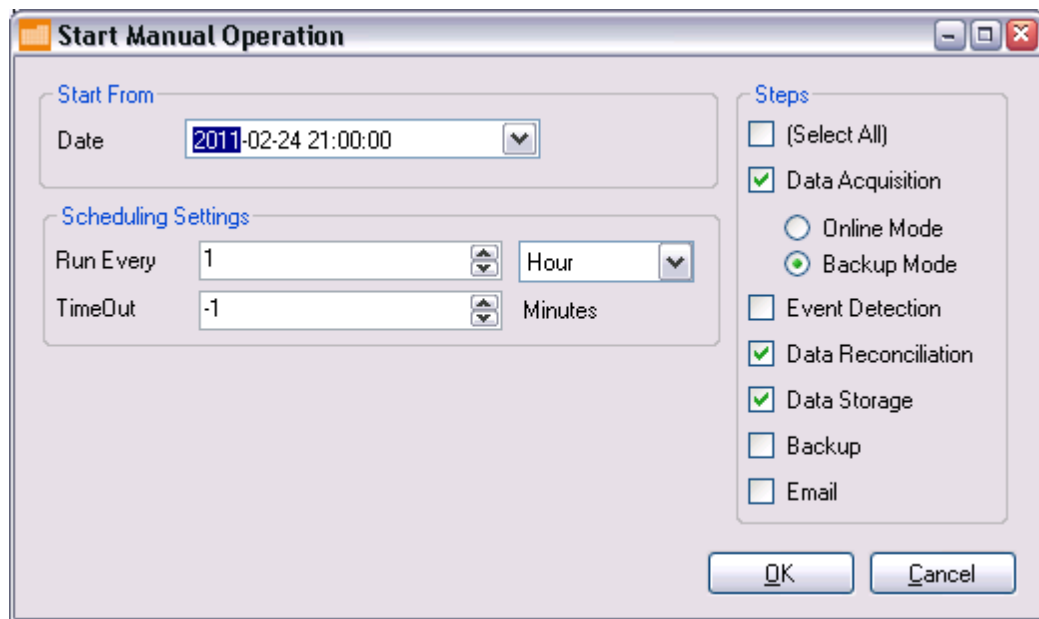


FIGURE 11 - START MANUAL OPERATION PAGE OF VALISCHEDULER

The execution of the tasks will start directly when you click on the "OK" button. The "Start Manual" button of the general page will be renamed "Stop Manual". Once a run is under processing, you cannot change the configuration settings. To stop the run execution, click on the button "Stop Manual". The button will be renamed "Stopping Manual".

Using ValiScheduler without ValiService, ValiScheduler will wait till the current tasks are finished.

Using ValiScheduler with ValiService, if the manual run has not yet been processed by ValiService, ValiScheduler will remove the run from the process queue of ValiService. Otherwise, ValiScheduler will wait till the current tasks are finished.

The "Start From" table is used to specify the date of the first run.

- Date: the date of the run to be processed

The "Schedule Settings" table is used to specify the settings of the automated procedure:

- Run Every: at which frequency successive runs will be scheduled
- Time Out: time period allowed for each task of the run. If the task lasts for a longer time period the task is cancelled. No time out will be applied if this value is negative. During the reconciliation step, ValiScheduler publishes a file next to the model file, called VALISTOP, once the timeout is reached. During resolution, ValiEngine will check for the existence of such a file. If it exists, ValiEngine will stop its resolution, publish the reports and close. If ValiEngine takes more than 2 minutes between the publishing of VALISTOP and its closing, ValiScheduler kills ValiEngine process and free its license (as well as the ValiFilter license)

The "Steps" table is used to specify the tasks that will be executed in the automated procedure:

- Data Acquisition: If selected, ValiScheduler will launch the retrieval of measurement data from DCS, PIMS, or other databases:
 - Online Mode: Start acquisition tasks defined in "Acquisition" page with "Online Mode" selected
 - Backup Mode: Typically used with "Rerun", start tasks defined in "Acquisition" page with "Backup Mode" selected and restore backup files (See "[backup](#)" page for more details about restoring backup files)
- Event Detection: if selected, ValiScheduler will launch the event detection (ValiEvent) on measurement data (See [ValiEvent](#) documentation)
- Data Reconciliation: if selected, ValiScheduler will launch ValiOnlineEngine
- Data Storage: if selected, ValiScheduler will launch the storing of results after the validation step
- Backup: if selected, ValiScheduler will back-up the selected files after the validation step
- Email: if selected, ValiScheduler will launch the automatic emailing according to the parameters specified in the "[Email](#)" page

The time window of the validation tasks is delimited by two dates: BeginDate and EndDate. The difference between both dates corresponds to the Time Interval:

$$\text{EndDate} = \text{BeginDate} + \text{Interval}[\text{IntervalDim}]$$

The Time Interval is equals to the time span value of "Run every".

The BeginDate will be fixed by the field "Date".

When executing a manual run, the application will be locked and will wait till the tasks are finished.

When using ValiScheduler coupled to ValiService, the tasks will be processed by the computer where ValiService is installed. If ValiService is already processing tasks, the manual run will be processed after all the tasks currently in processing.

3.3 ACQUISITION PAGE

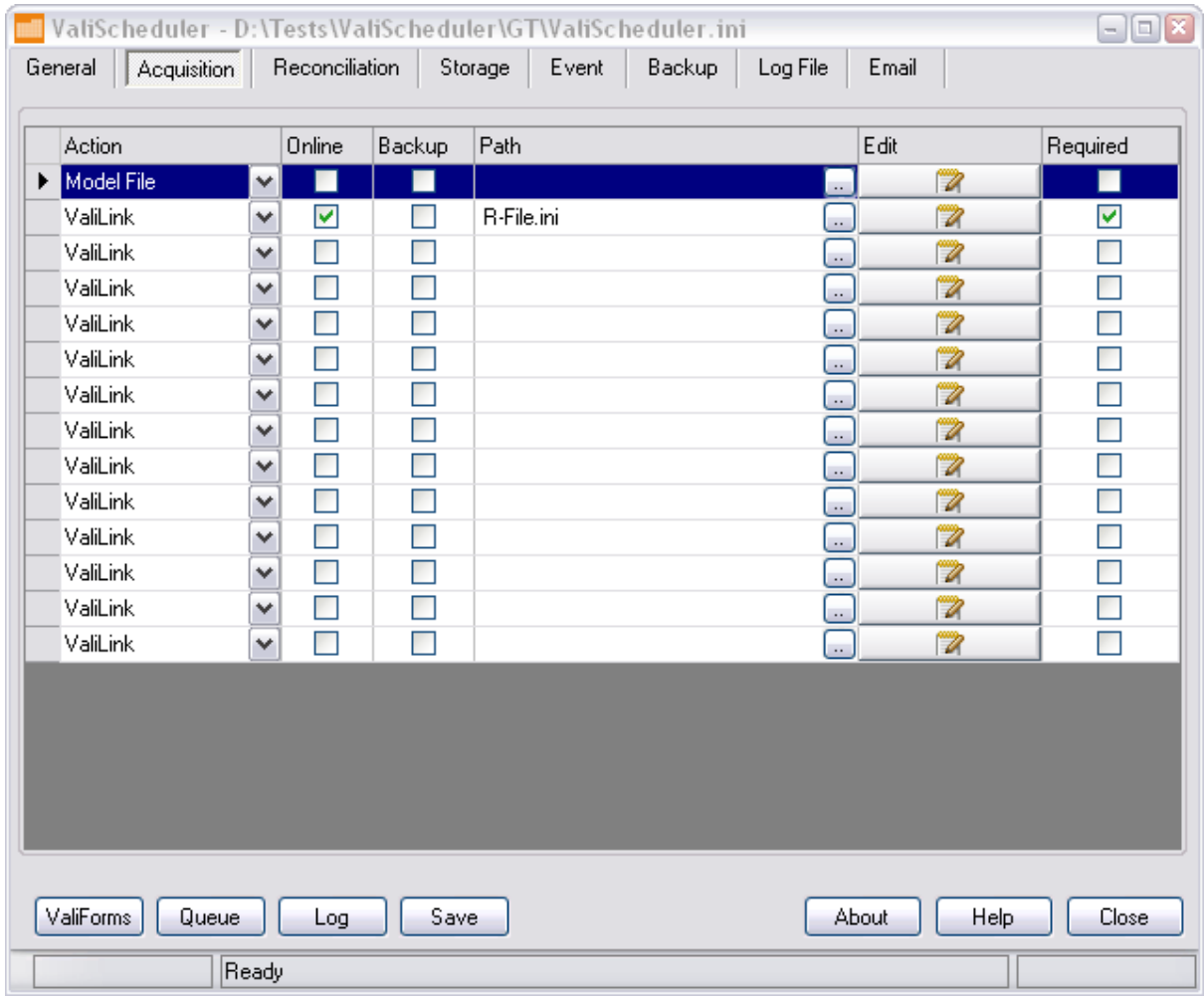


FIGURE 12 - ACQUISITION PAGE OF VALISCHEDULER

The Acquisition page is used to specify the options for getting measurement data from DCS, PIMS, or other databases.

The "acquisition" table is used to specify the acquisition settings. The first row allows you to launch ValiLink using the settings define in the Model file (through ValiModeller). For other rows, you can customize the action to execute:

- Action: the kind of action to execute:
 - ValiLink: Use ValiLink with a configuration file.
 - Batch File: Use a MS-DOS Batch file
 - ValiDBLoader: Use ValiDBLoader to store result in ValiDatabase.
- Online: if selected, the action will be executed for processing in "Online Mode". To specify "Online Mode", select "Online Mode" under "Acquisition" in the "Start Operation" page
- Backup: if selected, the action will be executed for processing in "Backup Mode". To specify "Backup Mode", select "Backup Mode" under "Acquisition" in the "Start Operation" page
- Path: specify the path and name of the ValiLink configuration file, batch file or ValiDBLoader configuration file
- Edit: allows you to edit the configuration file specified under "Path". Without ValiService, default editor will be started (notepad,...). With ValiService, a Belsim editor is used to retrieve content of the file through ValiService, modify it and send it back to ValiService. This allows to remotely editing the file installed on the application server
- Required: if checked, the result of this task is taken into account to determine the final status of acquisition step. According the status of acquisition, Data Reconciliation will be launched or not

For the syntax of initialization files of ValiLink, see chapter [ValiLink](#).

3.4 RECONCILIATION PAGE

The **Reconciliation** page contains a **VALI 4** tab and a **VALI 5** tab, respectively allowing you to launch the calculation using the VALI 4 Engine on a VALI 4 model or using the VALI 5 Engine on a VALI 5 application.

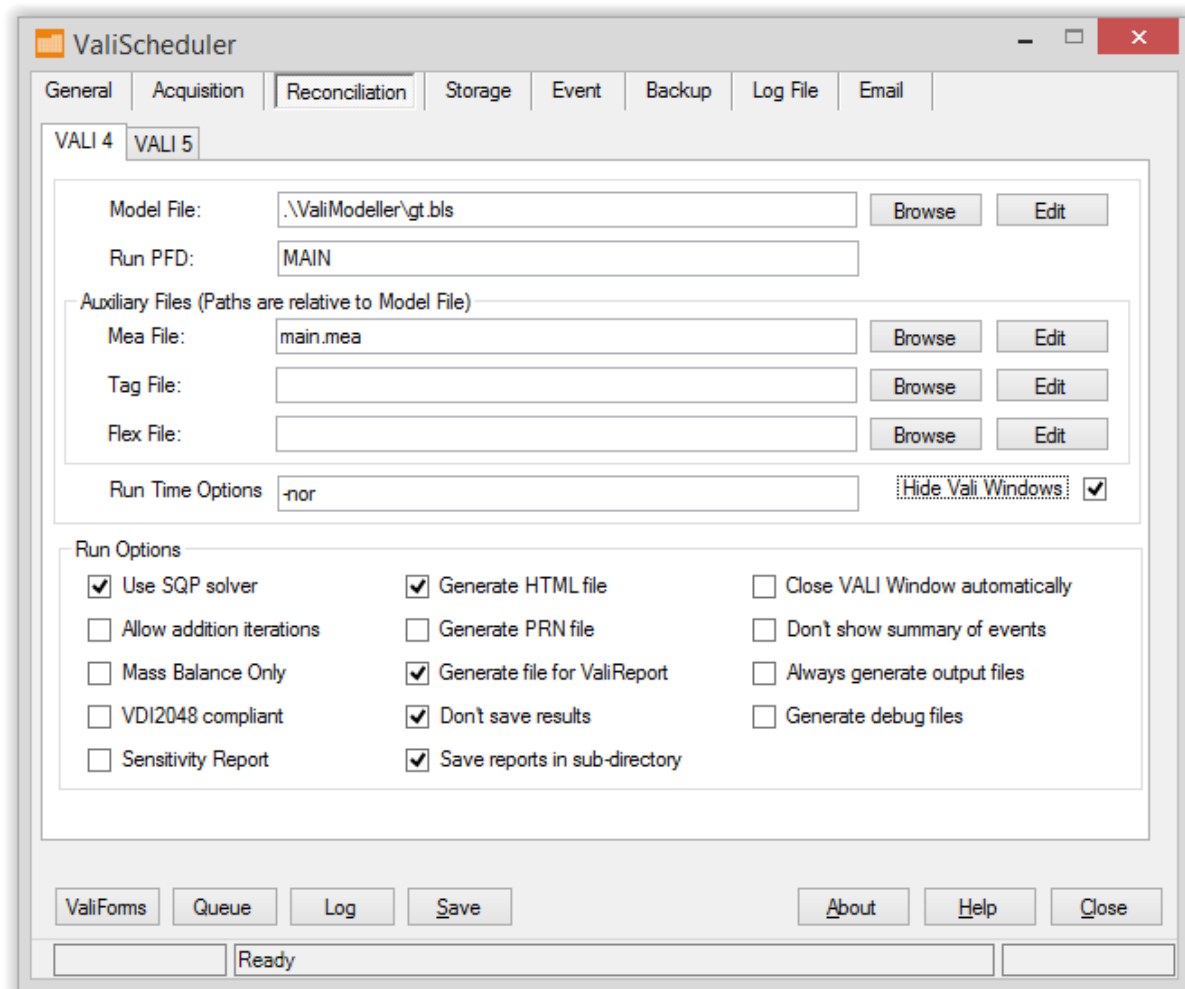


FIGURE 13 - VALI 4 RECONCILIATION PAGE OF VALISCHEDULER

In the **VALI 4 Reconciliation** page you specify the settings of ValiOnlineEngine (written in the pc.vif file):

- the name and location of the VALI model file
- which PFD of your model must be run
- the names of the auxiliary files (Measurement, Tag and Flex files)
- the runtime options (see section "[Running the model](#)")

When browsing auxiliary files, the path is relative to model file.

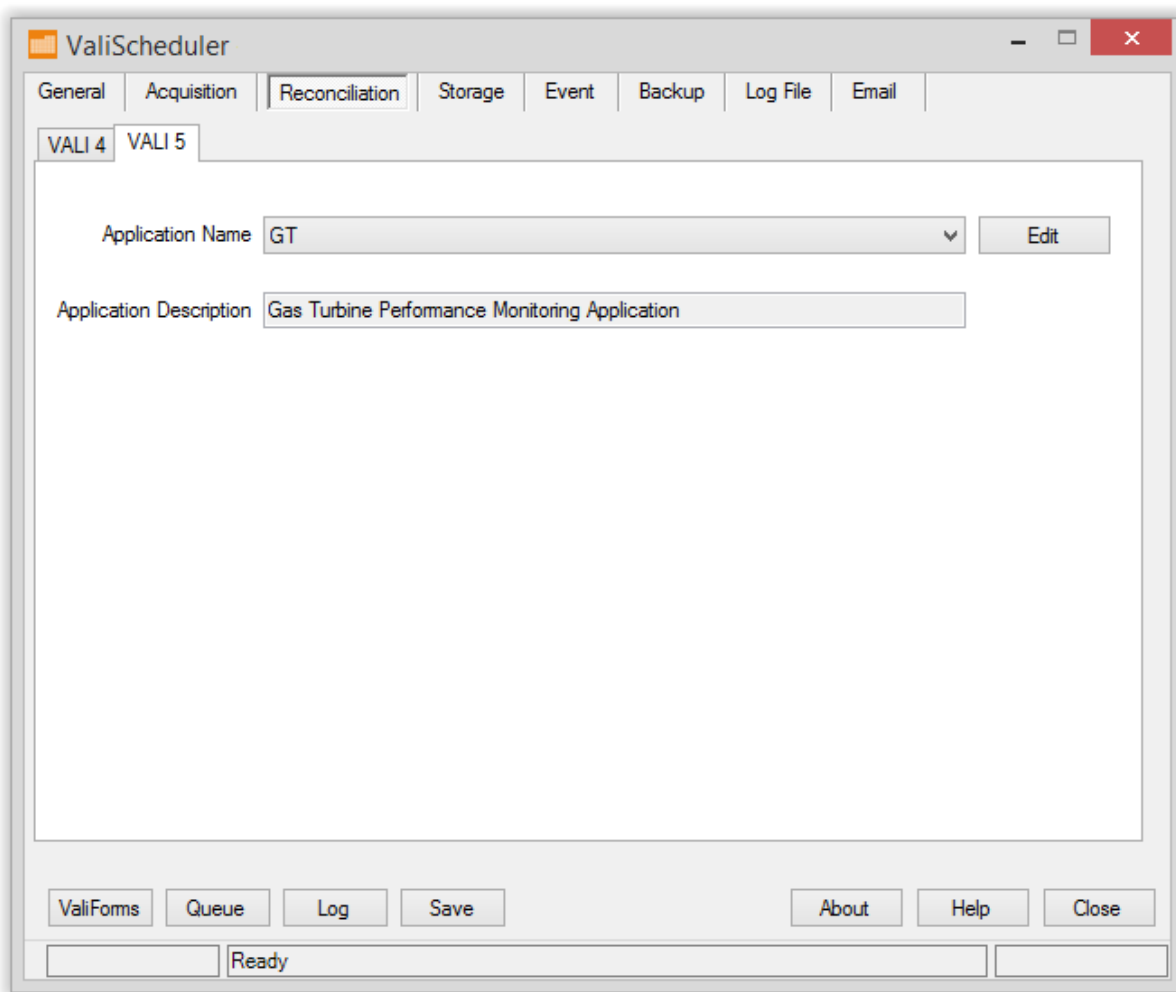


FIGURE 14 - VALI 5 RECONCILIATION PAGE OF VALISCHEDULER

In the **VALI 5 Reconciliation** page you select the VALI 5 Application on which you wish to run the calculation, from a drop-down list displaying all the applications defined in the VALI 5 workspace.

The **Edit** button allows you to edit the calculation settings of your application through VALI 5 ValiStudio.

3.5 STORAGE PAGE

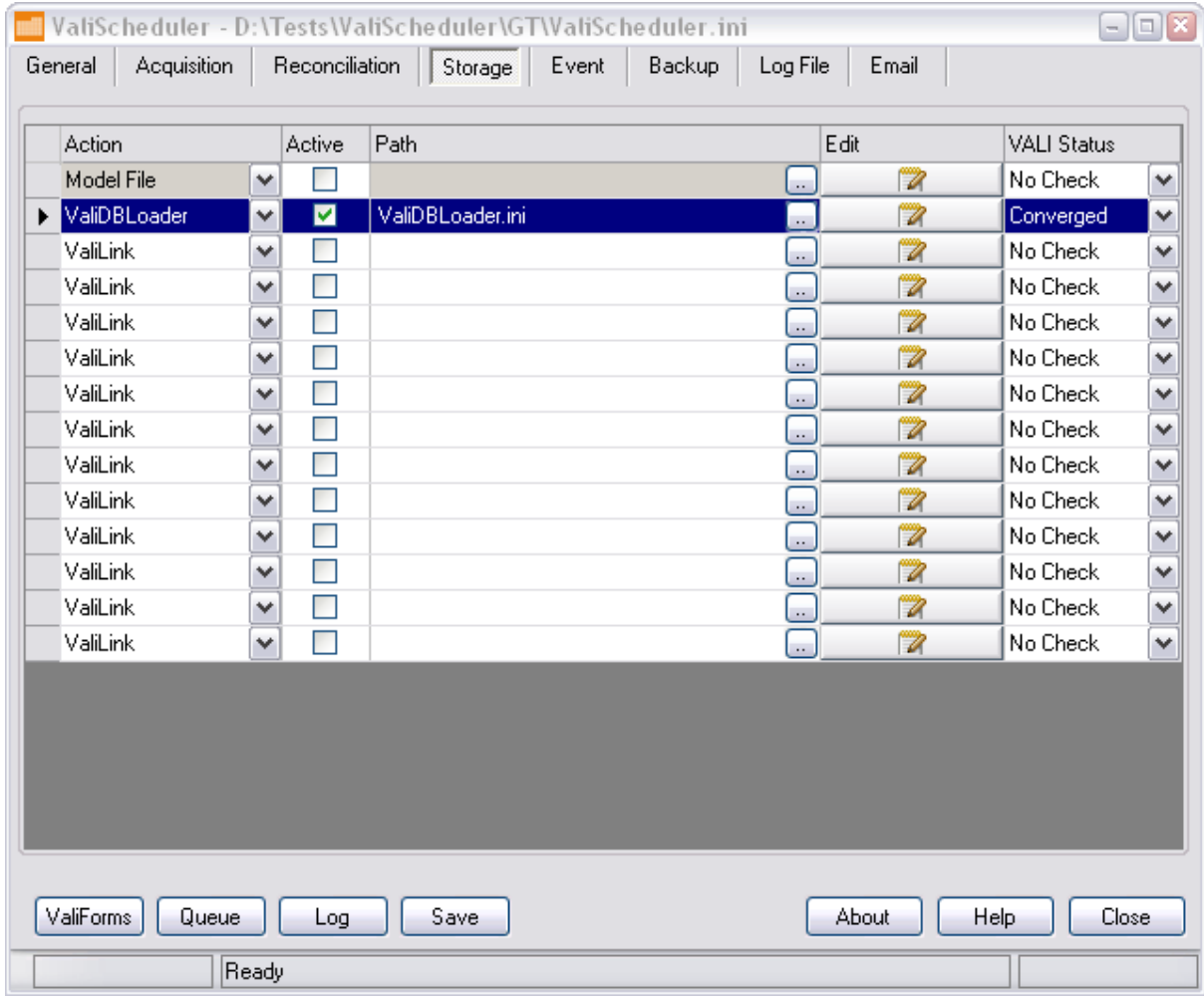


FIGURE 15 - STORAGE PAGE OF VALISCHEDULER

The **Storage** page is used to specify the options for storing results of the validation step.

The "Storage" table is used to specify the data storage settings. The first row allows you to launch ValiLink using the settings defined in the Model file (through ValiModeller). For other rows, you can customize the action to execute: ValiLink, MS-DOS Batch file, ValiDBLoader or ValiForms.

Any of these steps can be temporarily de-activated (Active check box) and executed or not according to the status of ValiEngine execution: No Check, Converged, No Gross Error, Failed.

For the syntax of initialization files of ValiLink, refer to [ValiLink file](#).

For the syntax of initialization files of ValiDBLoader, see chapter [ValiReport](#).

For the syntax of configuration files of ValiForms, see chapter [ValiForms](#).

3.6 EVENT PAGE

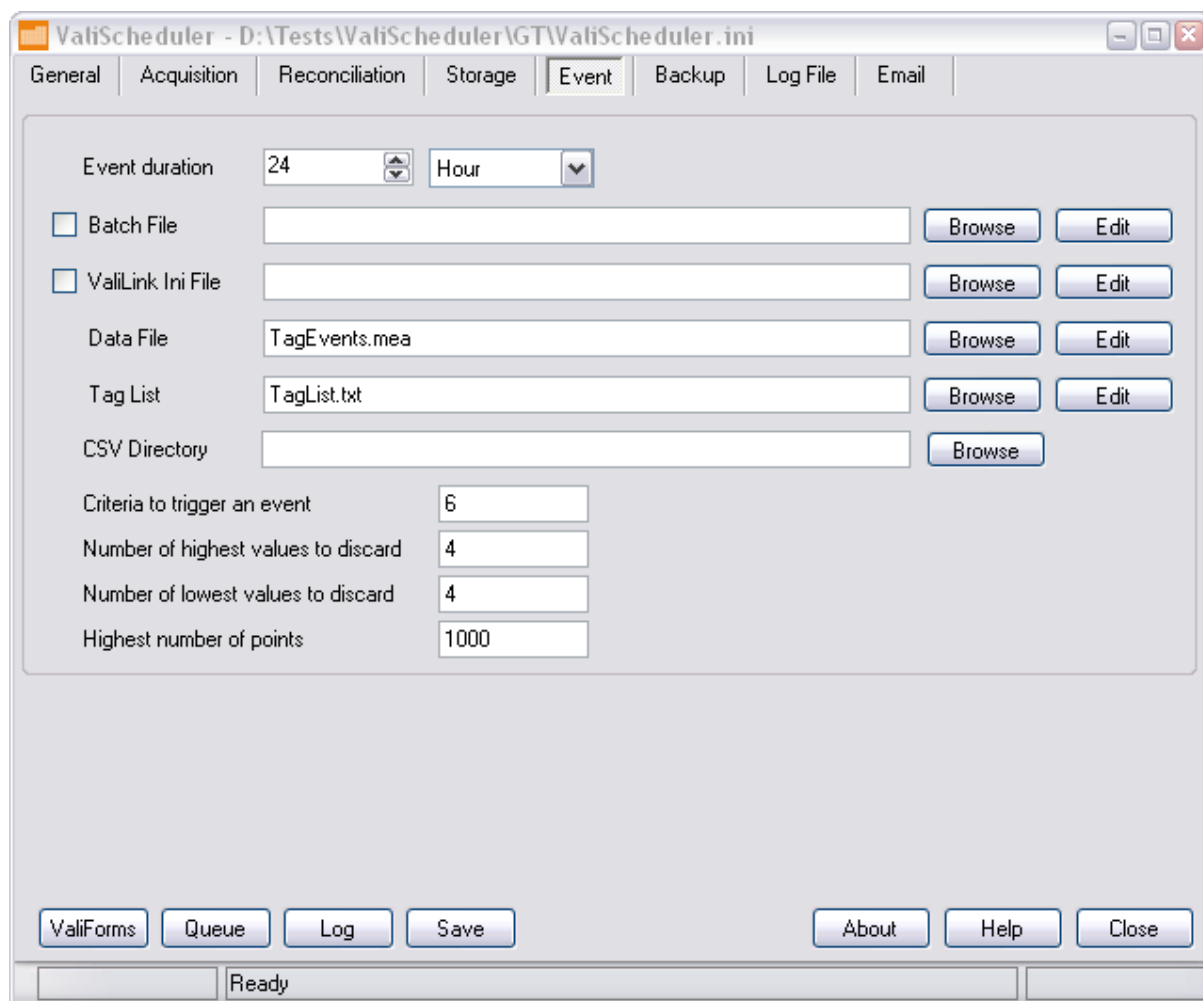


FIGURE 16 - EVENT PAGE OF VALISCHEDULER

ValiEvent is used to analyze measurement data and detect any sudden change in the gradient of one of those measurements. ValiEvent works generally with time intervals higher than the validation process itself in order to have a better insight of the plant operation (steady or unsteady) and detect efficiently measurement events.

The Event notebook allows to specify options for ValiLink in this particular automated process and options for ValiEvent itself:

- the frequency (**Event duration**) to be considered for this particular automated process
- the Batch File (if any)
- the ini files to be used by ValiLink to get the data for ValiEvent
- the Data File to be used as input file for ValiEvent
- the options for running ValiEvent (see ValiEvent documentation)

- the Output Folder for CSV file (output file of ValiEvent)

3.7 BACKUP PAGE

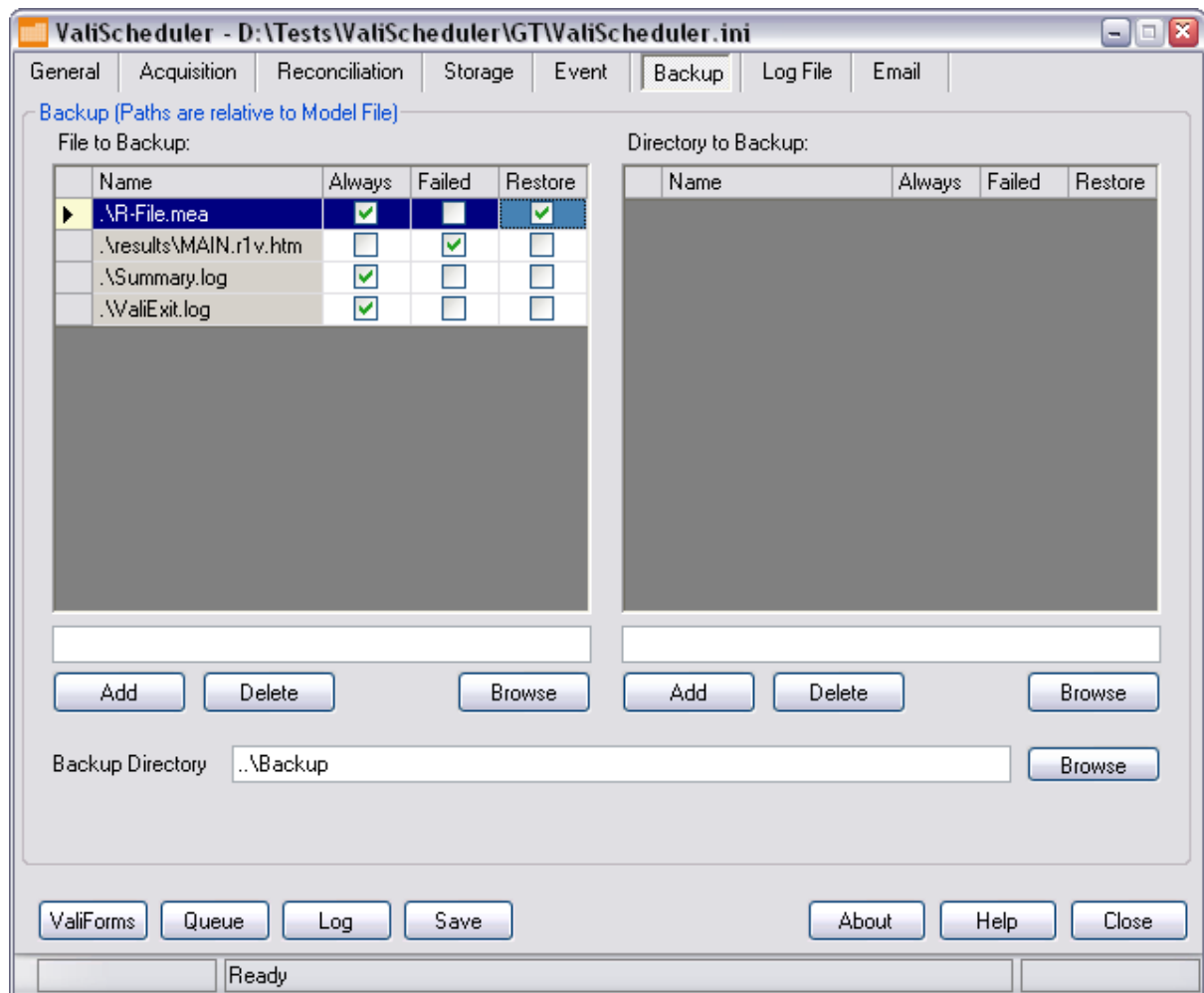


FIGURE 17 - BACKUP PAGE OF VALISCHEDULER

The Back-up procedure is used for example to have a more detailed information of the runs performed (Vali results files) or to be able to rerun special sets of data.

The settings enable to:

- select files or folders to be backed-up (using Add, Browse or Delete buttons)
- specify the Backup directory
- specify what to backup according reconciliation status (Always, Failed)

The "Restore" column is used to specify what files must be restored during acquisition phase when running in "Backup Mode".

The listed files are then copied into the Backup folder according as followed:

- <BackupDir>\<DateTime>\File
- <BackupDir>\<DateTime>\Folder

<DateTime> is the start date and time of the run under the format "yyyymmdd_hhmmss".

3.8 LOG FILE PAGE

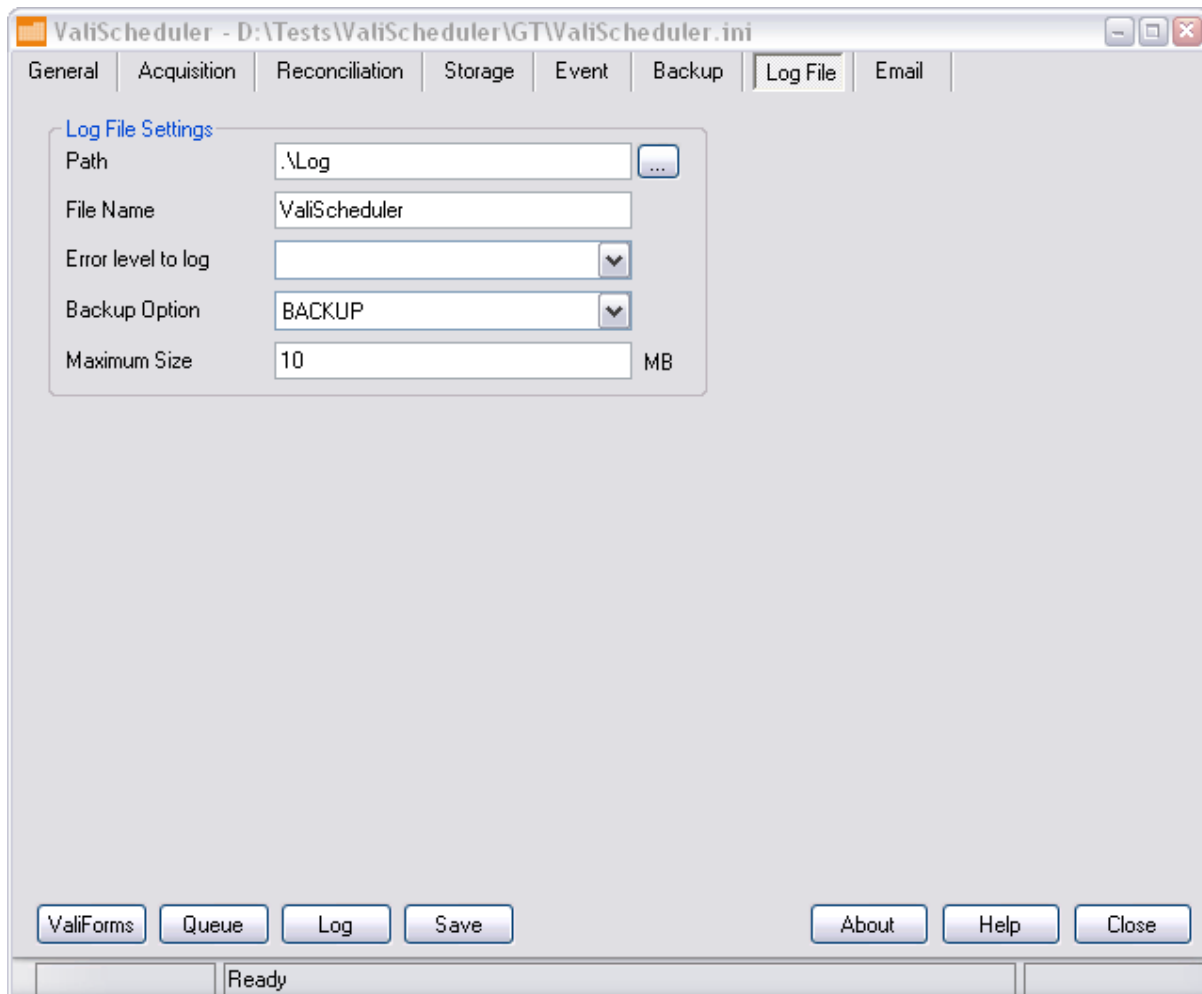


FIGURE 18 - LOG FILE PAGE OF VALISCHEDULER

The Log File page allows to specify:

- Path: path of the ValiScheduler log file,
- File Name : name of the log file (if left blank, the default name is ValiScheduler.log)
- Error Level to log: used to filter the level of information and error messages to be written in the log file. The log level can be one of the following: 0, INF, WRN, ERR, ERS, ERF, NONE, Only messages with levels higher than the log level are written. For instance, with ERR specified as log level, only ERR, ERS and ERF message will be printed. NONE means no message is logged, 0, means all message are logged
- Backup option:: ARCHIVE, BACKUP or APPEND:
 - BACKUP: old file ValisScheduler.log is renamed as ValiScheduler_bck.log, messages of the current run are logged into a new file ValiScheduler.log
 - ARCHIVE: messages are written in a new file Valischeduler_20031231_000000.log, where 20031231_000000 is the current date
 - APPEND: messages are logged into the old file ValiScheduler.log

3.9 EMAIL PAGE

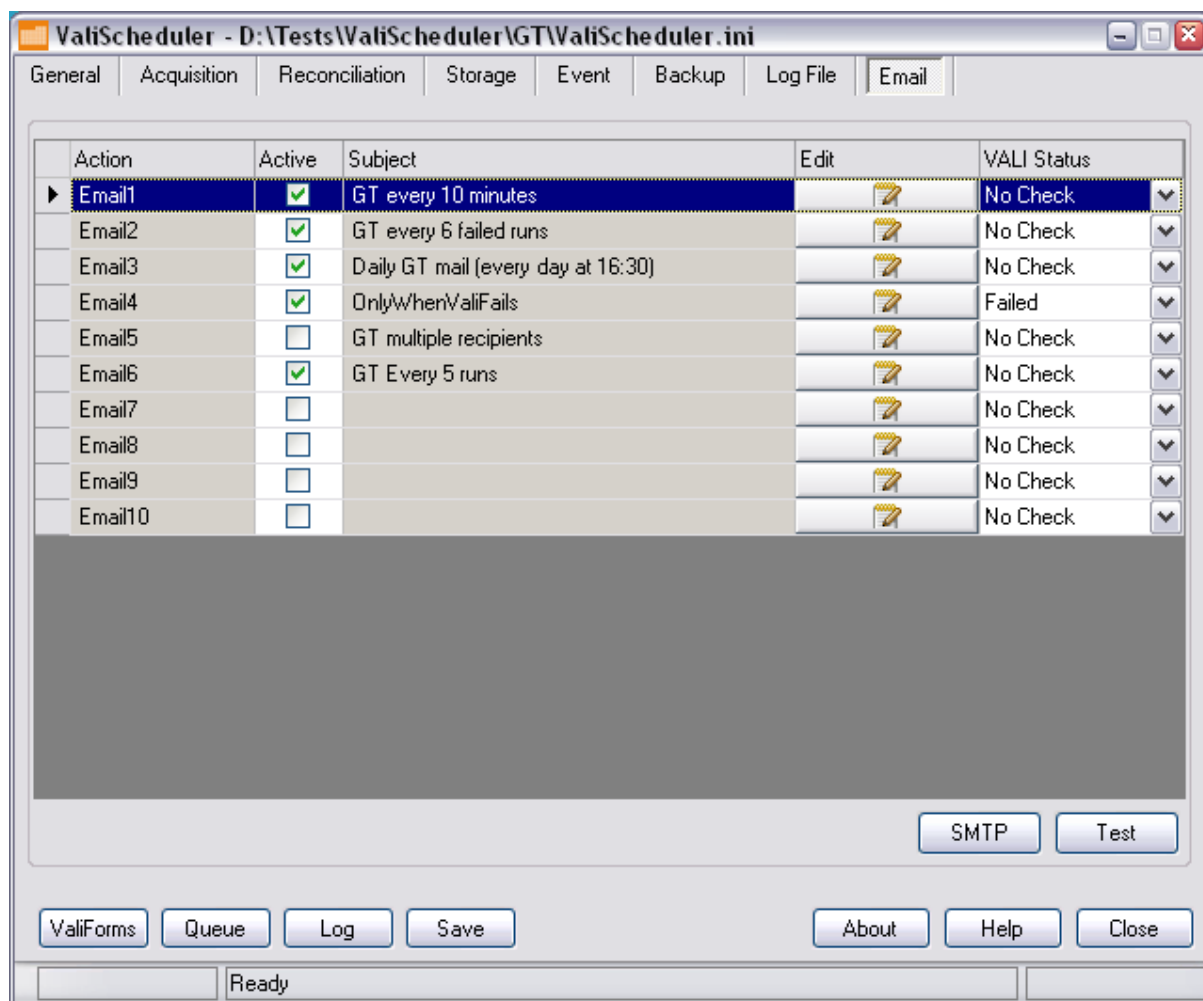


FIGURE 19 - EMAIL PAGE OF VALISCHEDULER

The Email tab allows to configure the automatic sending of e-mails according to predefined criteria. Up to 10 different e-mails can be sent, each based on different criteria. Clicking the Edit button  opens a new window for the configuration of an e-mail, where following parameters can be defined:

- the sender of the e-mail
- the recipient(s) of the e-mail; several recipients can be specified, and have to be separated by a semicolon (;)
- the subject of the e-mail
- the text of the e-mail (pre-defined keywords can be accessed using the Keywords button)
- the criterion to send the e-mails:
 - after a pre-defined number of failed runs
 - after a pre-defined number of runs
 - every day at a specified time
- files to be sent together with the e-mail; these files can be compressed in a zip file

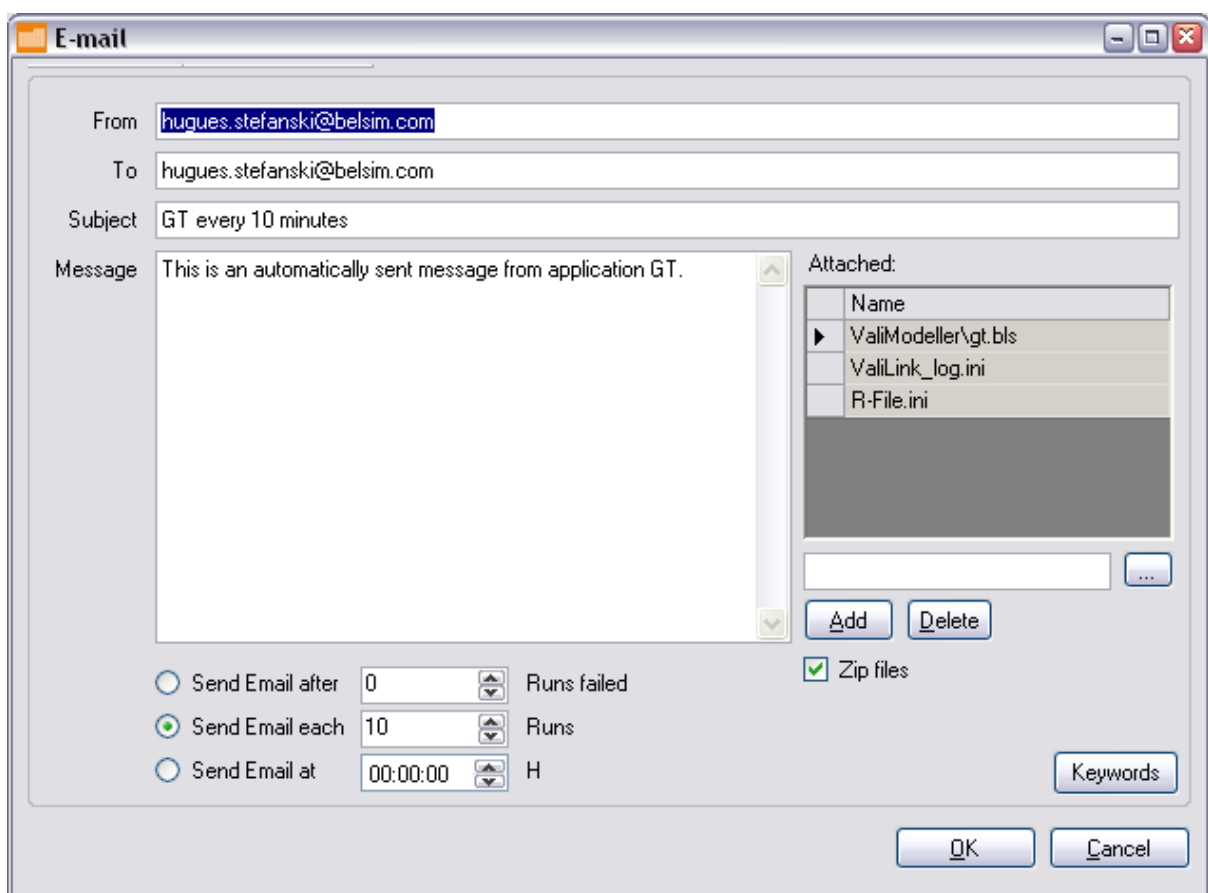


FIGURE 20 - EMAIL CREATION

SMTP button is used to configure the SMTP information required by the system to send emails. This button will open a configuration window:

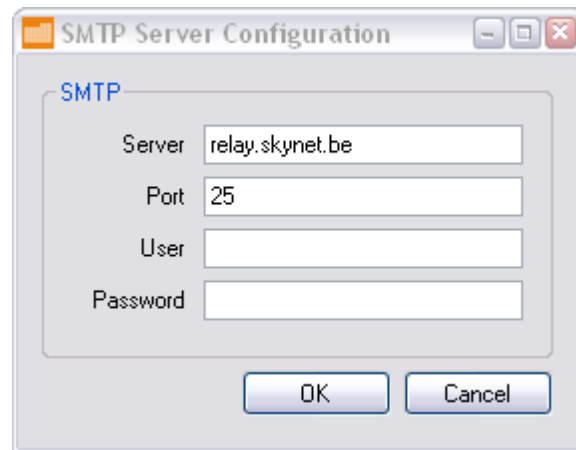


FIGURE 21 - SMTP SERVER CONFIGURATION PAGE OF VALISCHEDULER

Through this window, you can set:

- Server: address of the smtp server
- Port: the port that is used to send e-mails
- User: the user name that will be used to send e-mails
- Password: the password of the user used to send e-mails

The **Test** button allows you to test the configuration of each email, by sending selected e-mail.

SECTION 4: VALISERVICE

4.1 INTRODUCTION

ValiService is an application that will interact with ValiScheduler and ValiManager as a client-server network architecture.

ValiService is a NT Service (Win32 Application without graphical interface). It will act as a TCP/IP Server on the network and will be managed remotely through ValiScheduler and ValiManager.

ValiService will be used to process online validation with all the other tasks associated such as ValiLink, ValiDBLoader, ValiEvent, etc.

ValiService can be configured to restart automatically after a computer shutdown and will process all missing validation tasks to ensure no data is missing.

ValiService integrates security management, using Windows Active Directory, to control the access authentication to the working VALI directory. These security settings are used with ValiManager and are configured with LicenseManager (see LicenseManager for more information).

4.2 VALISERVICE ENGINE

ValiService can be configured to restart automatically with the computer it's installed on. When starting, ValiService will initialize a TCP/IP server and read a configuration file to fix internal settings. ValiService will then connect to the LicenseServer to get the list of validation application to process. In case of connection troubles (LicenseServer not started, network failure ...), ValiService will continue trying new connection till it succeeds. It will then initialize the "Processing Queue List" and retrieve the application settings from the LicenseServer. ValiService will then start its scheduler and every 5 seconds will check if there is a new validation task to process. The tasks to process are sent to the "Process Queue List" and ValiService continues its scheduling job.

The Scheduler engine can process several different applications. Stopping the real time run for one application will remove that application from the scheduling. It will also remove all the associated tasks from the "Process Queue List". Starting an online validation will add the application to the ValiService scheduling engine.

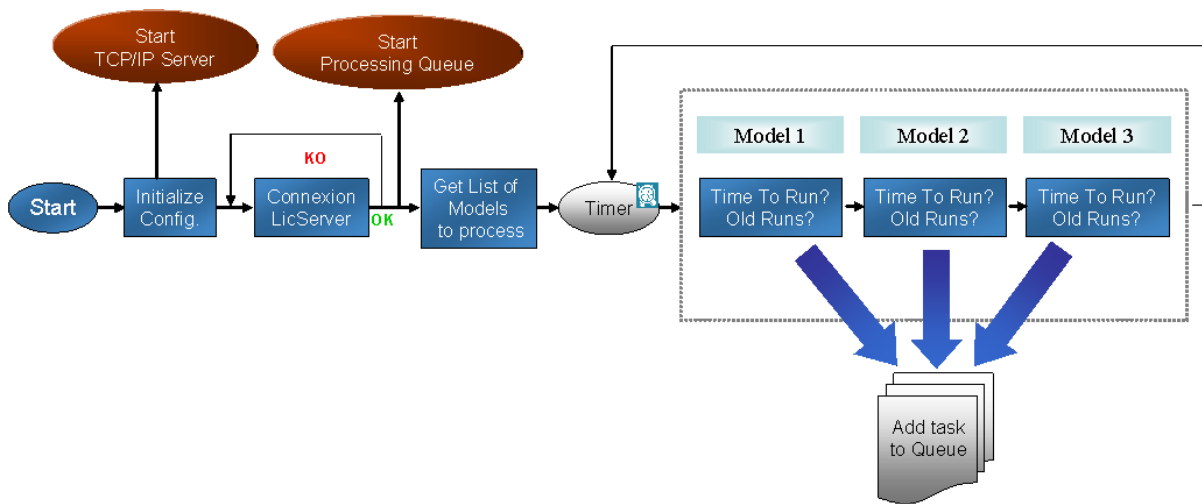


FIGURE 22 - VALISERVICE ENGINE DESCRIPTION

4.3 PROCESS QUEUE LIST

Whenever ValiService detect a validation application has to be processed, it will send all the required information to the "Process Queue List". Once a validation tasks is added to the "Process Queue List" it can not be modified. Stopping ValiService scheduling of one application will remove all the associated tasks from the "Process Queue List".

"Process Queue List" will check continually if a task has to be processed according to different priorities. He will process one task at a time. When the tasks are terminated, it will save the validation results then delete the task from the queue list. It can now process the next task in the queue list.

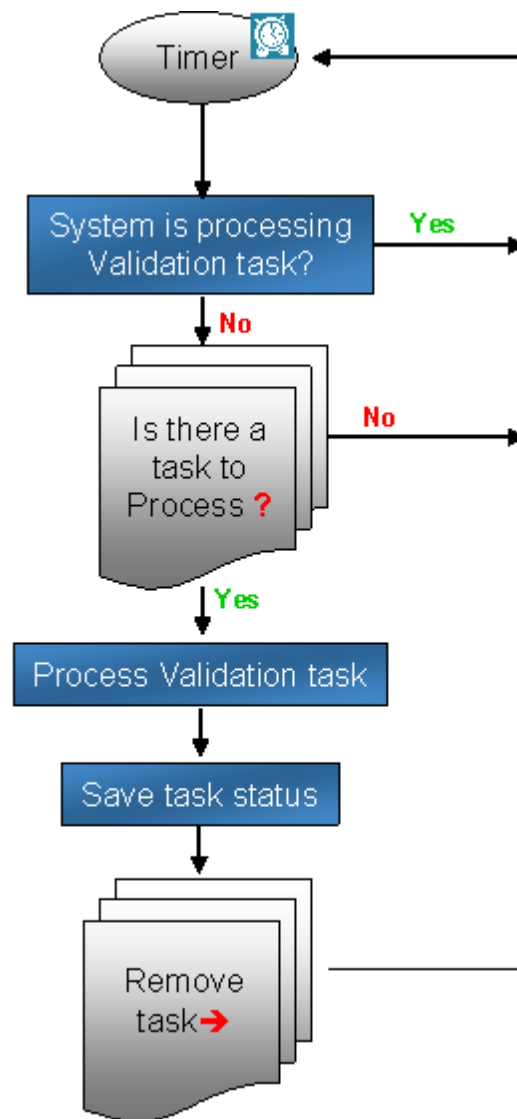


FIGURE 23 - PROCESS QUEUE LIST DESCRIPTION

How priorities are handled in the "Process Queue List"?

The notion of priority is based on three lists, three levels of priority. When ValiService needs to launch a task, this one is added to the corresponding list according to the type of the task (Manual, Real Time run,...).

When the "Process Queue list" needs to launch a task, it will look in the list of priority 1, if something has to be done. If this list is empty it will then look in the list of priority 2 and eventually in the lists of priority 3.

There will be as many lists of level 3 as ValiPerformance applications handled by ValiService. This enables load balancing between the reruns of applications, a run being picked up alternatively between the active lists.



FIGURE 24 - PROCESS QUEUE LIST PRIORITIES

4.4 CONFIGURATION OF VALISERVICE

To modify properties of ValiService:

- in the «Control Panel» of Windows, choose «Administrative Tools»
- choose «Services»
- in the opened window, select «Belsim ValiService»

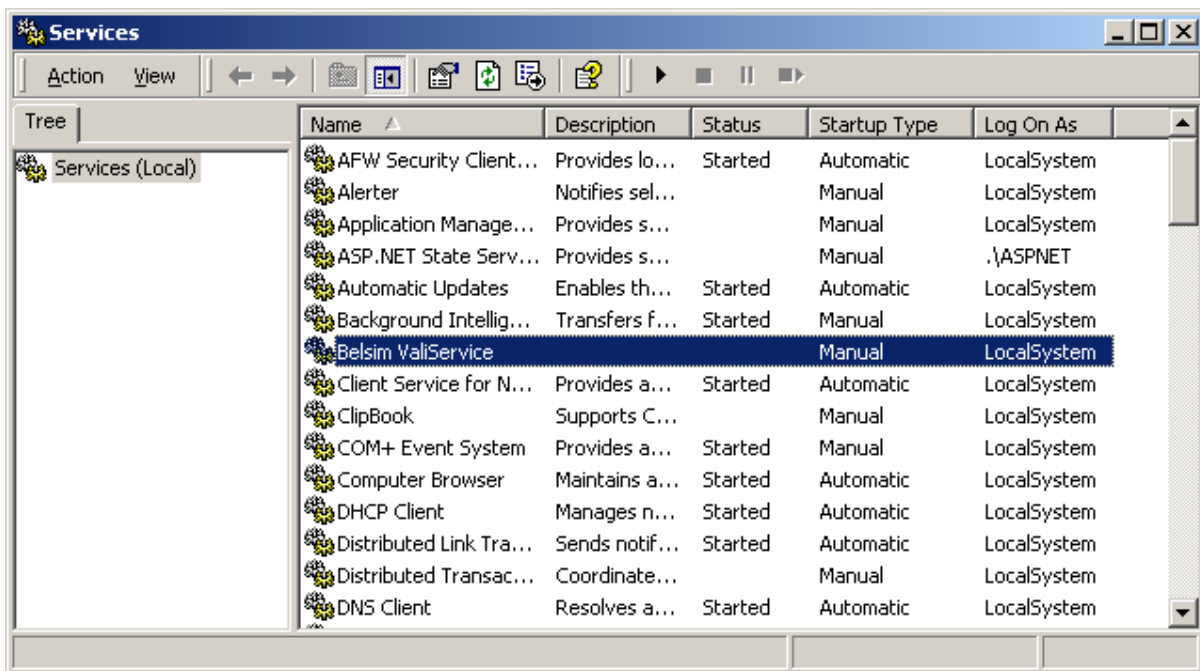


FIGURE 25 - SERVICES WINDOW

- to start the application, you just need to click in the arrow of the toolbar
- to access the properties of the Service, right click on «Belsim ValiService» and choose properties

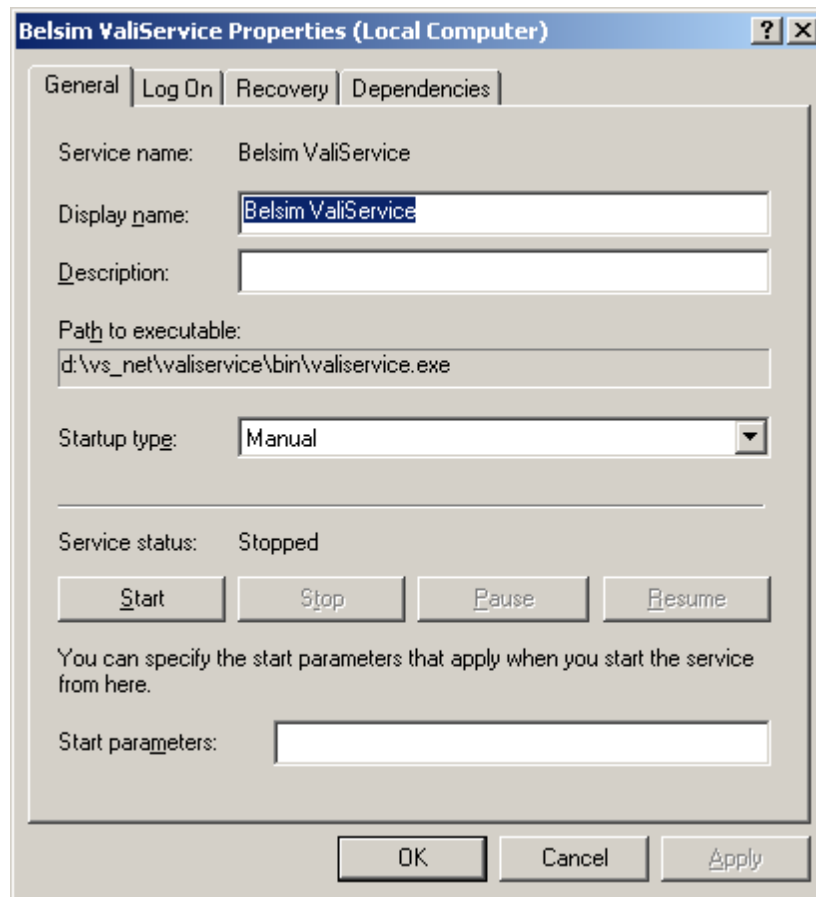


FIGURE 26 - CONFIGURATION OF SERVICE

ValiService generate a log file (ValiService.log) in the directory «App» of the Belsim installation. The Service acts as a TCP/IP server and uses the TCP port 4601 by default. It is possible to modify the port used by the server and the clients. To do that, execute the following steps on the server/clients:

- open Windows Explorer
- browse to the installation directory of VALI ([InstallDir] \Belsim \dat)
- edit the file "Belsim.ini"
- add under the section [others] the following line:
 ValiServicePort=XXXX where XXXX is the number of the TCP/IP port used
- Save the file

If the server uses a firewall, it is mandatory to open the port 4601 to allow connection between clients' applications and ValiService.

Please refer to LicenseManager help files to have more details on how to define an automatic runs processing using ValiService.



belsim

CHAPTER 16

VALIPRODUCTION

SECTION 1: INTRODUCTION

1.1 VALI SOFTWARE SUITE

VALI is a process oriented software aiming at determining and improving production efficiency, productivity and energy consumption while reducing production and maintenance costs.

With VALI, you move from data to information, from information to knowledge and finally from knowledge to action.

FROM DATA TO INFORMATION

Measurements are the essential source of information of VALI (in addition to the process topology and the process management information) and cover all information coming either from the plant (temperatures, pressures, rates, on-line analyses, etc.) or from the laboratory (GC, densities, etc.).

VALI converts the raw measurement data into reliable information by applying data validation technique involving among others heat and mass balances.

FROM INFORMATION TO KNOWLEDGE

From the coherent information, VALI determines accurate and reliable performance indicators for each equipment of the plant (heat rates, rotating equipment efficiencies, reaction's conversions, etc) as well as more global KPIs (Key Performance Indicators) like plant efficiency, plant specific product yields, etc.

VALI pinpoints instrumentation deviation and equipment degradation.

FROM KNOWLEDGE TO ACTION

By generating customised reports for each user (maintenance, control, management, production...), adequate action will rapidly be undertaken. Using validated data for generating warnings strongly reduces false alarms. A clear distinction is made between instrumentation problem and equipment degradation.

1.2 VALIPRODUCTION

ValiProduction monitors consumption of raw material, production and inventory of intermediate and final products. Using a rigorous and automatic procedure based on closed mass and material balances and statistical principles, ValiProduction provides a clear view of the production accounting of a full unit or a whole industrial site. The software allows exploiting as well closed heat balances and thermodynamic feasibility rules.

On a daily basis all information about production levels of each unit, inventories and receipts and shipments is reconciled providing a coherent site material balance. Thanks the functional redundancy provided by the plant model, the validated data set is highly accurate and reliable.

This coherent site material balances, obtained on a daily basis, leads to:

- accurate yield accounting
- reliable production report on inputs and outputs of each unit
- internal cost distribution per unit
- quick identification of imbalances improves loss control
- accurate and coherent data import in ERP systems, a must for correct working of ERP systems

1.3 INFORMATION FLOW

ValiProduction consolidates, on a daily basis, the material balances of the different production units existing on a given site or even on several interconnected sites. These material balances can be performed either:

- **on a global mass balance basis** : measured mass (or volume) flow rates, in terms of tons (or m3) entering and tons (or m3) going out of each production unit, are reconciled to generate a coherent mass balance of the whole site. This approach is typically applied in refineries and covers the whole site including the tank farm
- **on a chemical compound basis** : additional information is then required on the composition of the various streams and the reactions schemes. This approach is typically applied in chemical and petrochemical plants

In both cases, ValiProduction will be fed with data about the shipments and receipts, raw data about inventories (feedstock, intermediate and final products) and about the production levels.

For yet better performance and profitability, ValiProduction can also be fed with validated data coming from ValiPerformance applications (see below synergy between ValiPerformance and ValiProduction).

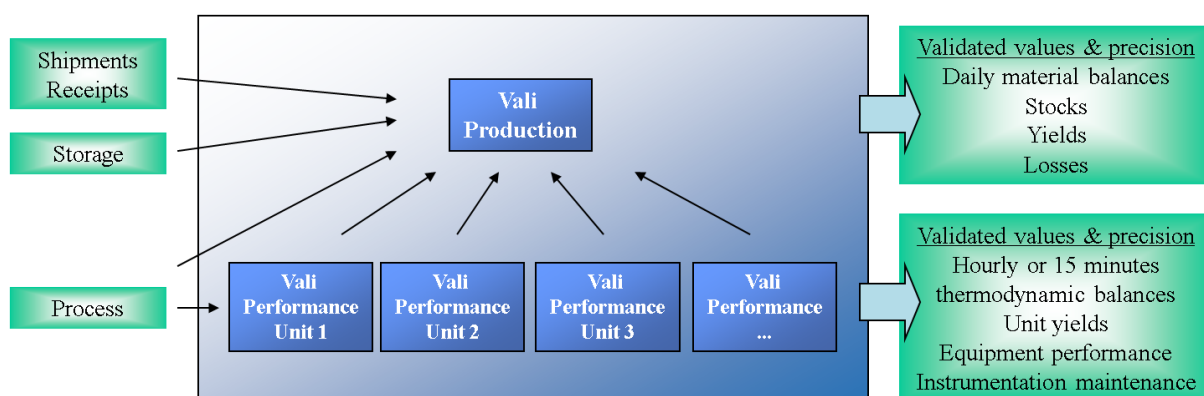


FIGURE 1- VALIPRODUCTION ENVIRONMENT

The validated production rates, inventories and yields are the most accurate values that can be obtained using the existing data. The so-produced information corresponds to the most probable plant operation because the set of corrected values respect coherent (closed) balances (mass, compounds and sometimes heat and thermodynamic balances).

1.4 SYNERGY BETWEEN VALIPRODUCTION AND VALIPERFORMANCE

ValiPerformance is a software solution for detailed process performance monitoring of process units: most often implemented online and based on consistent mass, heat and thermodynamic balances, this product computes and reports on the key performance indicators of a process such as process yields, reaction selectivity, delta_T approach to the equilibrium, energy consumption, equipment efficiency (e.g. heat transfer coefficient, turbine and compressor efficiency, etc.).

Based on the same data validation technology as ValiProduction, it uses more detailed models and provides more insight about the internal operation of each production unit. Running on a different time schedule (typically each hour or each 15 min) it takes advantage of the additional information coming from detailed energy and thermodynamic balance to produce performance indicators of unequalled quality in terms of reliability and accuracy.

By feeding ValiProduction with validated values from ValiPerformance applications, one benefits at the global level of this high quality data, making ValiProduction results even more reliable and accurate.

1.5 VALIPRODUCTION INTEGRATION

By essence, ValiProduction application interacts with other elements of the enterprise. A number of modules are available to ease the integration and provide the users with a convenient yet efficient set of tools for the deployment and maintenance of the system.

The access to the different parts of the application is performed through ValiManager application which provides the necessary framework for multi-users usage with adequate security measures, model versioning, operations logging, etc.

Authorized users maintain the production model using an efficient graphical user interface ValiModeller.

Interfacing with external systems (ERP, SQL database, PIMS, LIMS and DCS systems) is performed through a generic application ValiLink which provides different "plugs" for each system to be connected. Manual input can also be taken into account, through customized interfaces.

Data provided by ValiLink is then filtered giving an adequate set of input measurement data to ValiEngine so that convergence rate of the computation engine is maximized.

After validation through the ValiEngine, results are, on one side, sent to external systems using again ValiLink interfacing application and, on the other side, stored in ValiDatabase for further processing like analysis, reporting or consolidation.

Customs reports are easily created through the ValiReport extension of Microsoft Excel.

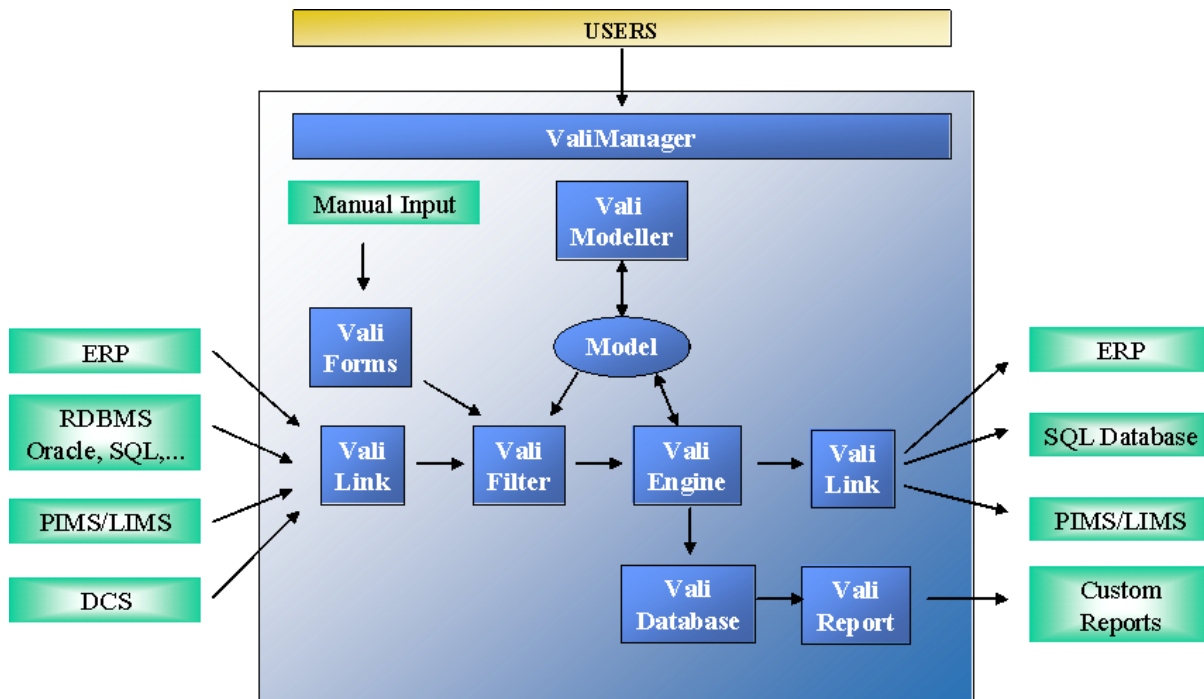


FIGURE 2 - A TYPICAL VALI PRODUCTION ENVIRONMENT

1.6 VALIPRODUCTION WORK-FLOW

The work-flow of each application is different and takes into account the following elements:

- multiple users: different users can be involved in the work flow with different roles:
 - entering or verifying input data
 - introducing manual corrections to improve results quality
 - verifying validation results and possibly introducing further corrections
 - approving the validation results
- multiple phases and steps: various tasks have to be performed:
 - acquiring data, either automatically from a data historian or through manual input
 - validating the data
 - exporting validated results to external systems
 - producing production reports
- data synchronization issues:
 - input data are not available at the same time
 - input data are not available at the same frequency

The work-flow is defined in the ValiDatabase, according to the concepts of phases and steps.

Phases are used to logically associate a number of steps and have a status (ON HOLD, IN PROGRESS, COMPLETED) derived from the status of their steps.

The steps are the core elements of the work-flow and correspond to the individual actions that must be performed to establish the production balance. The steps have various properties:

- a frequency of execution: daily, weekly or on specific week or month days
- conditions of processing that must be met before the task is executed: availability of data, completion of other steps in the same daily run or in other runs, conditions on date and time
- control access specifying which users can interact with the step
- detailed description of the task to be performed, including configuration files, report files and log files
- a status according to the execution procedure

Depending on the type of action to be performed and to configuration settings, different statuses are used.

For an automatic step, that is a step without direct user interaction during the execution, the following statuses are used:

-  ON HOLD: the conditions of the steps are not yet satisfied
-  QUEUED: the conditions are now satisfied, the action has been queued
-  IN PROGRESS: the action is currently executed
-  DONE: the step has been successfully executed
-  WARNING: the step has been executed and reported a warning status
-  FAILED: the execution failed

After the step execution, various user actions can be performed according to the execution status.

If the step was successfully executed ( DONE) or reported some warning ( WARNING), the following options are available:

- the step is automatically set as being approved
- designated users are required to approve the results: they can then choose to approve the results, to rerun the step after some corrections or (depending on the configuration) to skip the step

If the step failed ( FAILED), the designated users will have the options to rerun the step after some corrections or (depending on the configuration) to skip the step.

To be considered as completed, the final status of an automatic step must be either:

-  APPROVED: the step has been either automatically or manually approved
-  SKIPPED: the step has been manually skipped

For a manual step, where users have to enter data, the following statuses are used:

-  ON HOLD: the conditions of the steps are not yet satisfied
-  QUEUED: the conditions are now satisfied, the action has been queued
-  IN PROGRESS: the action is currently executed
-  APPROVED: the step has been either automatically or manually approved
-  SKIPPED: the step has been manually skipped

1.7 VALIMANAGER APPLICATION

ValiManager is designed as the primary portal for the follow-up of the ValiProduction work-flow. It provides the following features:

- user profile: Master, Operator, Modeller, etc. According to the user profiles, different access rights will be granted: model creation and modification, input of manual data, manual correction of raw data, approval of validated results, etc.
- traceability and versioning: model changes are documented and the evolution of the model is monitored so that old data can be reprocessed with their corresponding model
- environments: training and development environment are available avoiding any interaction with the official production environment. Export of data to external systems is only possible through the official production system
- task scheduling: authorized users determine when the validation runs are performed. They verify the relevance of validation results, introduce manual corrections where necessary and finally approve the results and make them official
- system maintenance: comprehensive log files enable to quickly trace failures of the system, like interfacing problems, networking problems, unavailability of critical data, inadequate operation modes, etc.
- reporting: various reports are issued and can be accessed by the users according to their needs: production data, inventories, suspect sensors for instrumentation maintenance, etc.

SECTION 2: CONFIGURING A VALIPRODUCTION APPLICATION

2.1 A FLEXIBLE CLIENT/SERVER SCHEME

ValiProduction is a client/server system that is based on four main components.

- ValiManager: client process through which users will use and maintain the ValiProduction application. Several ValiManager client sessions can be used at the same time on the same ValiProduction application
- a ValiProduction Server: the ValiService application processes automatically daily runs and gives access (through ValiManager Client) in a controlled way through the various functionalities of the ValiProduction application
- a ValiDatabase: an SQL Database contains the configuration settings of the application, the status of the daily runs and the validation results
- Belsim LicenseServer: server process that handles security related settings like user rights or license allocation

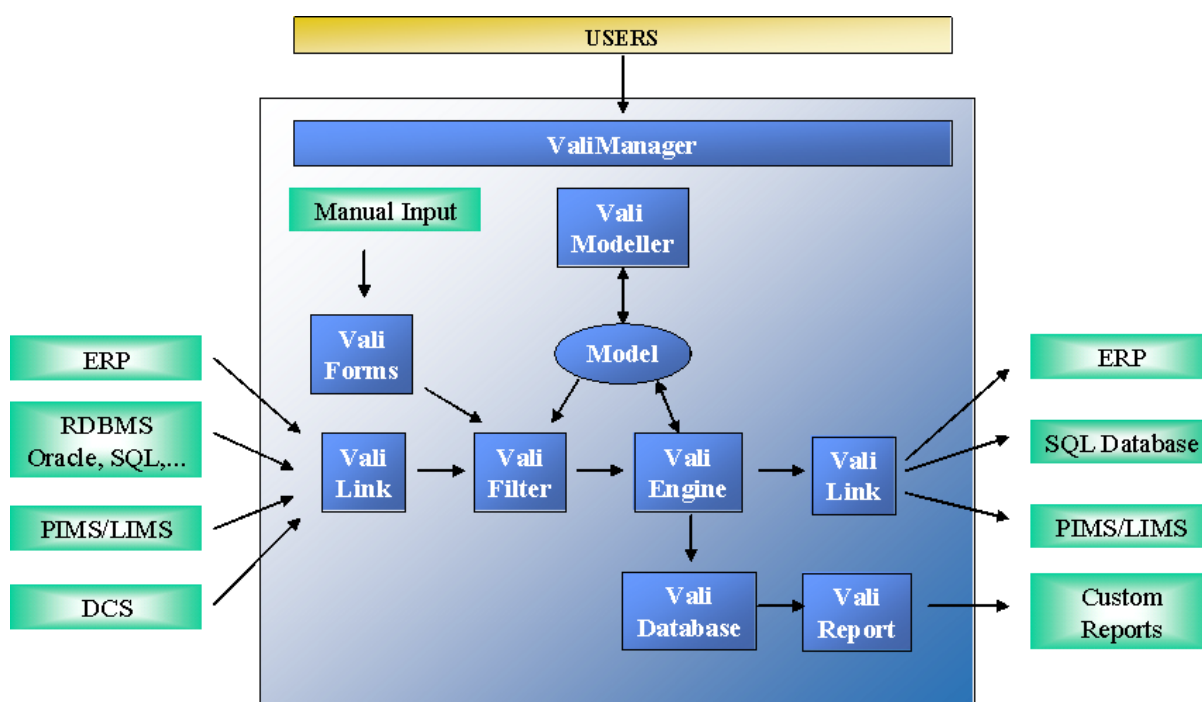


FIGURE 3 - VALIPRODUCTION CLIENT/SERVER ARCHITECTURE

Each of these four components can be installed on different computers or on a single computer.

Refer to the "Installation Procedure" for practical details of installation strategy and procedure.

2.2 USER RIGHTS

The user rights are defined through Belsim LicenseManager/Server according to a number of roles:

- Master: gives full access to the application including runs approval and changes to official model but not to configuration panel neither to ValiModeller
- Operator: enables to enter manual input data and manual corrections on official runs (using for example ValiForms) and to ask for reruns of steps
- Configurator: gives access to configuration panels
- Modeller: enables to work in manual mode but not to save as Official model or Official run
- Reader: can only see results (overview of processed runs and access to ValiReport)

Additionally, specific user rights, based on user names, can be specified when configuring the various steps of the application.

2.3 APPLICATION DIRECTORY STRUCTURE AND PROCESSING

DIRECTORY STRUCTURE

Each ValiProduction application has its own root directory, which must be accessible through a unique path from ValiService on the ValiProduction server as well as from each ValiManager client. In particular, if a network drive is used, it must be defined on all computers that need to access the application.

The following folders are used (MyApplic standing for the ValiProduction application name):

- ...\\MYAPPLIC\\: root directory of the application
- ...\\MYAPPLIC\\OFFICIAL\\: official files (bls, mea, etc) of the application
- ...\\MYAPPLIC\\ARCHIVES\\YYYYMMDD_HHMMSS\\: back-up of old Official files (bls, mea, etc) of the application
- ...\\MYAPPLIC\\RUNSAUTO\\YYYYMMDD_HHMMSS\\: files of the application copied run per run
- ...\\MYAPPLIC\\VALIREPORT\\: ValiReport files

AUTOMATIC PROCESSING

According to a user-defined schedule, the runs are automatically processed according to the following procedure:

- at a selected time of the day, new runs are created when necessary. Typically, a new run is created each day for the preceding day. In case the ValiService has been shutdown for some days, the missing days will be automatically created. The application files are automatically copied from the **Official** folder to the run directory in the **RunsAuto** folder and all steps of the new runs are set to ON HOLD.
- each step of the workflow is scanned on a regular basis to analyse if some work must be performed, according to steps status and conditions.

2.4 CREATING A NEW VALIPRODUCTION APPLICATION

To create a new ValiProduction, you must proceed through the following steps on the ValiProduction server:

- install VALI software (including ValiService) and reboot the server,(see installation procedure)
- create a ValiDatabase (MS SQL required) using ValiDatabaseCreator application
- create a ValiManager application using Belsim License server setting it OFF and set-up user rights. The ValiProduction application folders will be automatically created on the server by ValiService
- copy the files of the application (model files, measurement files, configuration files for ValiLink, etc) in the Official folder of the ValiProduction application folder
- configure the application workflow in ValiDatabase using the MS-SQL Enterprise Manager (Tables Steps_Definition, Steps_Conditions, Steps_Interaction, Phases_Definition)
- activate the ValiProduction application using Belsim License Manager (setting it ON)

Steps 2 and 3 are further detailed below.

Step 5 is described in the section "[Configuring the workflow](#)".

CREATING A VALIDATABASE (USING VALIDATABASECREATOR)

If you wish to use a MS-SQL server database, you will need to create the SQL database using ValiDatabaseCreator application (accessible from Vali 4 Windows menu).

Create ValiDatabase

SQL Server Authentication

IP/UNC:

User name:

Password:

ValiDatabase

☒ Create DataBase & Tables
☐ Create Tables

DB name:

☐ ValiProduction Database

Create Plant ☐

LicenseServer:

Plant name:

Create SQL User ☐

Login:

Password:

Confirm Password:

FIGURE 27 - CREATING A SQL SERVER VALIDATABASE

- SQL server authentication (mandatory):
 - IP/UNC: IP/UNC of the SQL server
 - User name: login name of administrator account used to create the database
 - Password: password of the administrator account
- ValiDataBase (mandatory):
 - Create Database & Tables: a new Database must be created at the same time
 - Create Tables: only the tables need to be created
 - DB Name: name of the SQL database
 - ValiProduction : tables needed for ValiProduction have to be created
- Create Plant (optional): must be checked if you want to create a new plant in Belsim license server in order to access the database. If not checked, it is assumed you will use a plant already defined in Belsim license server:
 - LicenseServer : IP / UNC of Belsim LicenseServer
 - Plant Name: name of the plant to be created
- Create SQL User check-box: if checked the (Login) user will be created in the SQL server:
 - Login: user name to be used to connect to the SQL database
 - Password: corresponding password

- Confirm password: password confirmation

The **Show SQL Script** button shows the SQL script code that needs to be executed. This enables to execute the corresponding script at the SQL server management tool level.

The Plant can also be defined through Belsim LicenseManager under the **Plant** tab:

Field	Value	Description
Plant Name	SQLPLANT	Plant name
Plant Owner	ADMIN	Any user name defined in License Server
Server Type	MS-SQL	Mandatory
Database Log Name	sa	SQL user name
Database Password	xxxxxx	Corresponding password
Confirm Password	xxxxxx	
Server IP or UNC	SQLServer01	SQL server IP or UNC
Database Name	ValiDatabase	Database name

Buttons: OK, Cancel

FIGURE 28 - CONFIGURING AN SQL SERVER PLANT

CREATING A VALIMANAGER (VALIPRODUCTION) APPLICATION

Each ValiProduction application must be defined at Belsim LicenseServer level, through Belsim LicenseManager.

For each ValiProduction license that you have you can define up to two ValiProduction applications:

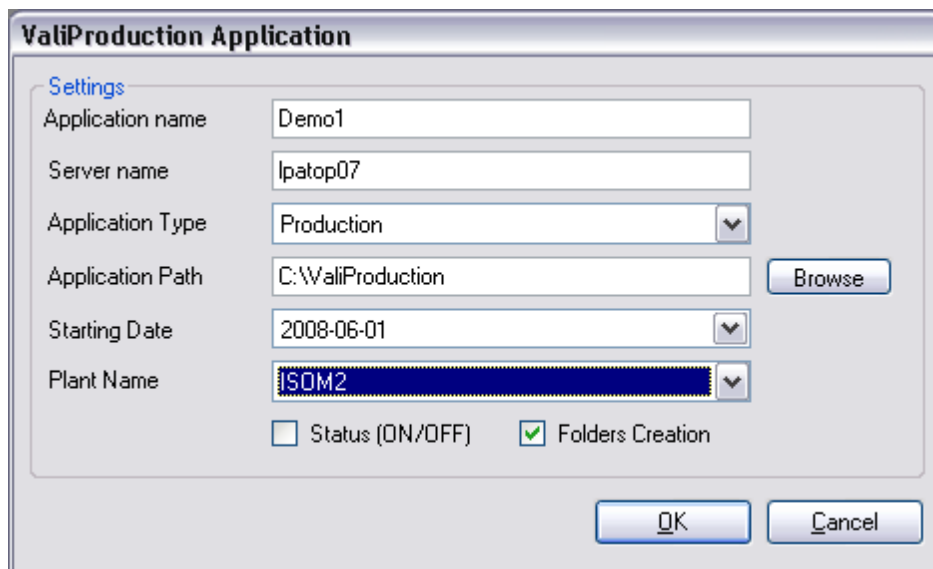
- one in "Production" mode, which is the official ValiProduction application
- one in "Test" mode, which enables to define and use a ValiProduction application without any interaction with the official application, for example for testing or for training purpose

Attention:

1. A "Test" application will never return any result to external systems (PIMS, databases).
2. You can also set both applications in "Test" mode, but you can never have both applications in "Production" mode.

To create a new ValiProduction you must have the ValiProduction Admin rights (at Belsim LicenseServer level):

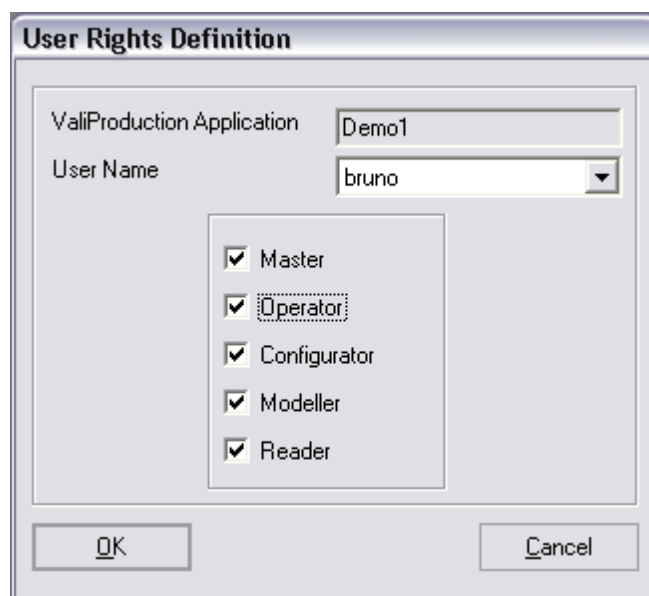
- start-up Belsim LicenseManager
- go to the ValiManager page
- press **Add Application** button
- fill-in the properties of the application



The screenshot shows the 'ValiProduction Application' dialog box. It has a title bar with the text 'ValiProduction Application'. Inside, there is a 'Settings' section with several fields: 'Application name' (text box with 'Demo1'), 'Server name' (text box with 'lpatop07'), 'Application Type' (dropdown menu with 'Production'), 'Application Path' (text box with 'C:\ValiProduction' and a 'Browse' button), 'Starting Date' (dropdown menu with '2008-06-01'), and 'Plant Name' (dropdown menu with 'SOM2'). At the bottom of the settings section, there are two checkboxes: 'Status (ON/OFF)' (unchecked) and 'Folders Creation' (checked). At the bottom of the dialog box, there are 'OK' and 'Cancel' buttons.

FIGURE 4 - CREATING A VALIPRODUCTION APPLICATION

- set-up user rights by adding authorised users using the Add_User button and specify their role:



The screenshot shows the 'User Rights Definition' dialog box. It has a title bar with the text 'User Rights Definition'. Inside, there is a 'ValiProduction Application' dropdown menu with 'Demo1' selected, and a 'User Name' dropdown menu with 'bruno' selected. Below these, there is a list of roles with checkboxes: 'Master' (checked), 'Operator' (checked), 'Configurator' (checked), 'Modeller' (checked), and 'Reader' (checked). At the bottom of the dialog box, there are 'OK' and 'Cancel' buttons.

FIGURE 5 - ADDING AUTHORIZED USERS

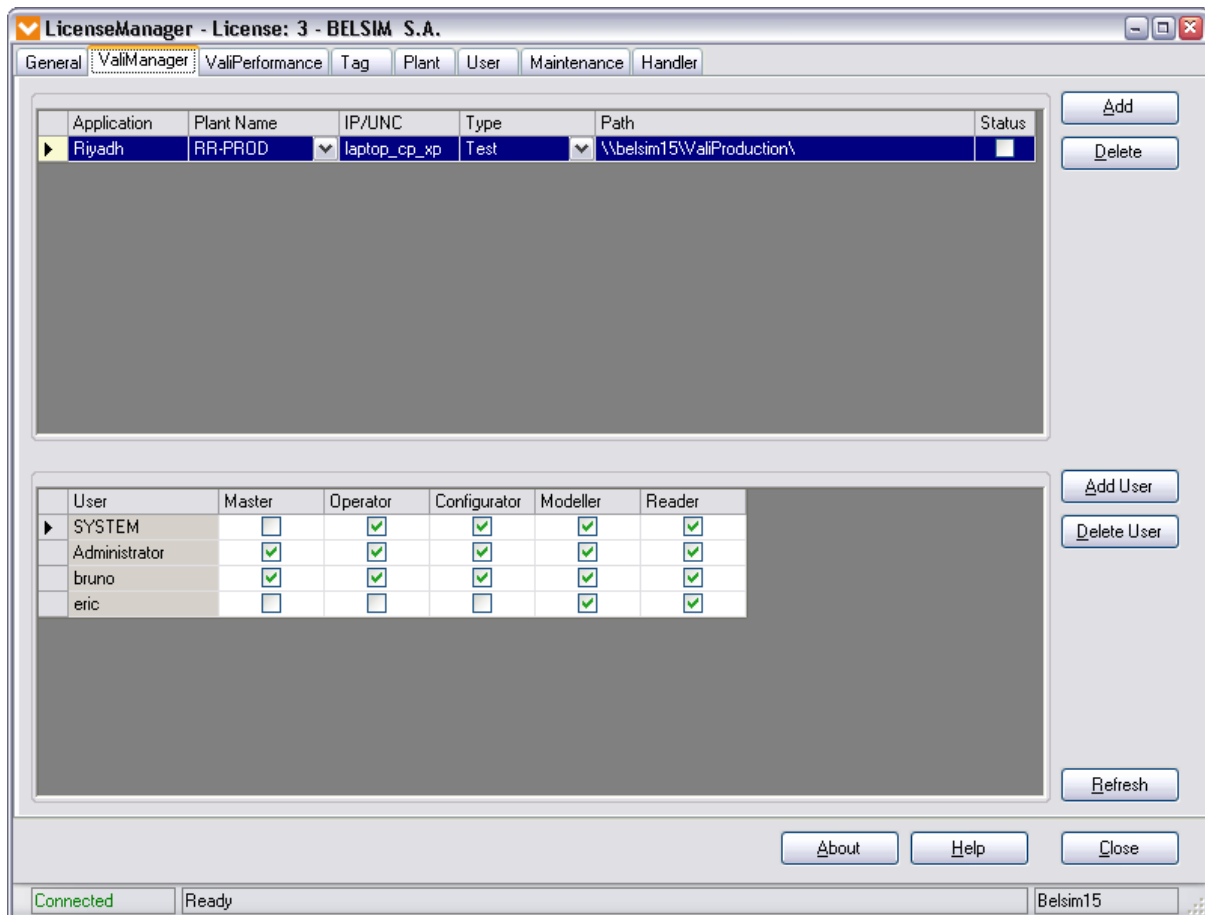


FIGURE 6 - PROPERTIES OF VALIPRODUCTION APPLICATIONS

2.5 CONFIGURING THE WORKFLOW

DEFINING THE STEPS: STEPS_DEFINITION TABLE

The Steps_Definition table has the following fields:

- Step_ID: ID of the step
- Phase_ID: ID of the phase to which the step belongs
- Short_Description: step description (32 chars)
- Long_Description: step description (128 chars)

Each step has a unique ID, a short and a long description, and belongs to a phase:

- NextPosition: order of the step: the next position defines the order of the steps in the execution sequence
- Next_Enabled(1/0): when set to 0 the step will not be considered. This setting indicates a step that is not used any more and must therefore not be considered in the work-flow
- Action_ID: definition of the step task

The Action_ID defines the action to be executed during the step (see Table below).

Depending on the action, some files used by the task must be defined in the Action_File. The Log_file is the name of the logging file created by the task, enabling authorized users to review the task execution while the Results field indicates the name of any result file created by the task.

The **Send Mail** ini file must contain two sections Message and EMAIL (replace words in italics by relevant information, use %RUNDATE% to include the run date):

```
[Message]
From=myname@mycompany.com
To=someone@mycompany.com
CC=
Subject=Information about run %RUNDATE%
Content File=Somefile.txt

[EMAIL]
Server=SMTPSERVER
Port=25
User=
Password=
```

The **Copy Last** ini file must contain one section FILES:

```
[FILES]
From=filetobecopied
To=targetfile
Overwrite=1 or 0
Log=name of log file
```

ACTION_ID	DESCRIPTION	ACTION_FILE	LOG_FILE	RESULT FILE
1	no action at all			
2	N/A			
3	ValiLink Input	ini file of ValiLink	Log file of ValiLink	mea file
4	ValiEngine (Valiauto.exe)	vif file	r1v.htm file	htm file
5	ValiDBLoader	ini file [mea file]		
6	ValiEvent	Arguments of ValiEvent		valievent.htm

ACTION_ID	DESCRIPTION	ACTION_FILE	LOG_FILE	RESULT FILE
7	Batch file (.bat)	Batch file name and arguments	Log file (if any)	Result file (if any)
8	ValiForms	-	xml file mea file	mea file
9	N/A			
10	N/A			
11	ValiLink Output	Ini file of ValiLink	Log file of ValiLink	
12	Send Mail	Ini File	Log file	
13	ValiLink Input (bls)	Model file (bls)	Log file of ValiLink	Mea file
14	ValLink Output (bls)	Model file (bls)	Log file of ValiLink	
15	Copy Last: copy a file from preceding run	Ini File	Log File	
16	Restore Backup	-	-	-
17	ValiForm Background Mode	xml file	-	-
18	ValiEngine V5	The name of the application to process, and optionally, the name of the Calculation Scenario to use. The 2 parameters shall be separated by a space character.	r1v.htm file	htm file

TABLE 1 - DEFINITION OF THE ACTIONS

- Action_File: file(s) used by the step
- Log_File: log file of the step
- Results: Name of the result file

- User_type: type of users in charge of the step (0: no user defined, 1: user defined by role, 2: user defined by name)
- User_name: Role or User Names, depending on User_type
 - Roles: Master=0, Operator=1, Config=2, Modeller=3, Reader=4. Several roles can be entered, and have to be separated by a |
 - Names: any user name. Several names can be entered, and have to be separated by a |
 - if no user is defined, then only the Master can approve, skip or re-run the step
- Rerun_at_step: in case of re-run, the procedure is restarted at the corresponding step:
 - 0 (by default) means the current step
 - a positive value refers to a step by its ID
 - a negative value refers to a step relatively to the current step (-1 means the preceding step)

All steps between the re-run step and the current step are set back to ON HOLD.

- Time_Out: Time-out after which an automatic step is cancelled

Should a time-out occurred, the step status is set to FAILED.

- Can_Skip (0/1): the step can be skipped by the user
- Auto_Complete (0/1): the step is automatically completed if the step executed successfully
- Auto_Complete_Wrn (0/1): the step is automatically completed if the step executed with warning status
- Optional (0/1): the step is optional

If a step is optional, its status is not used to determine whether a phase has been completed or not.

- Test (0/1): step not processed in a Test environment

When the ValiProduction application is in Test mode, the step is not executed. The step is automatically set to DONE once its conditions are satisfied (for a manual input step) or when reaching the In Progress status (for an automatic processing step).

- Schedule: defines on which days the step is to be processed
- Schedule_Bits: bits settings further defining these days

SCHEDULE	DESCRIPTION	SCHEDULE_BITS
0	Daily processing	not used

1	Week days	The first 7 bits are used to select the days (bit 1 is Monday)
2	Month days	The first 31 bits are used to select the days of the month. (bit 32 indicates the last day of the month)

TABLE 2 - DEFINITION OF THE STEP SCHEDULE

- Duration: ValiEvent alternate duration (in minutes)

Used by ValiEvent actions to select a duration different then the default run duration.

SYNCHRONISING THE STEPS: STEPS_CONDITIONS TABLE

The Steps_conditions table specifies the conditions to be met for a step to be executed. If several conditions are set on a given step, then all conditions must be satisfied for the step to execute. The Steps_conditions table has the following fields:

- Condition_ID: unique ID of the condition
- Step_ID: ID of the step on which the condition applies
- Short_Description: short description (32 chars)
- Long_Description: long description (128 chars)

Usually the **Short_Description** is used to describe the step while the **Long_Description** is used to describe the condition.

- Condition_Type: integer
- Condition_s1: further description of the condition (file name)
- Condition_i1: ID of the step that must be completed
- Condition_i2: ID of the run
- Condition_time: time delay (minutes)

CONDITION_T YPE	CONDITION _S1	CONDITION_I1	CONDITION_I2	CONDITION_TI ME
1: File existence	Name of the file			
2: Step completion		>0 ID of step <0 relative step ID	<0 relative run ID	

3: Time elapsed				Elapsed time since the run end time

TABLE 3 - DEFINITION OF THE STEP CONDITIONS

DEFINING THE INTERACTION WITH VALIDATABASE: STEPS_INTERACTION TABLE

The Steps_Interaction table defines changes that must be made to ValiDatabase when a step is set to a particular status. For example, a particular information message can be written in one of the INFOi fields of the RUNS table when a particular step is completed. The Steps_Interaction table has the following fields:

- ID: unique ID of interaction
- Short_Description: short description (32 chars)
- Long_Description: Long description (128 chars)

The interaction is defined by a unique ID, a short and a long description.

- StepID: ID of the step
- Status: status of the step

These fields indicate that an interaction must be made when the step Step_ID is set (manually or automatically) to the given Status.

STATUS	DESCRIPTION	STATUS	DESCRIPTION
1	ON HOLD	6	WARNING
2	PENDING	7	FAILED
3	QUEUED	8	COMPLETED
4	IN PROGRESS	9	SKIPPED
5	DONE		

TABLE 4 - THE STATUS IDS

- Type: type of interaction
- Tag: name of the tag to be updated
- Value: value to be given to the tag

TYPE	DESCRIPTION	TYPE	DESCRIPTION
1	INFOi	6	Penalty
2	Measured value	7	Status
3	Measured accuracy	8	Used value
4	Validated value	9	Used accuracy
5	Validated accuracy		

TABEL 5 - INTERACTION TYPES

GROUPING THE STEPS INTO PHASES: PHASES_DEFINITION TABLE

Phases are simply a way to group logically linked steps. The phase to which a step belongs is defined in the Steps_Definition table. The Phases_Definition table has the following fields:

- Phase_ID: unique ID of the phase
- Short_Description: short description (32 chars)
- Long_Description: long description (128 chars)

A phase is described by an ID, a short and a long description.

The status of the phase is linked to the status of the steps that are part of the phase:

- ON HOLD: all the steps are still ON HOLD
- COMPLETED: all the steps (except the ones specified as optional) are completed or skipped
- IN PROGRESS: in all other cases

2.6 TRACK CHANGES

STEPS INTERACTION

To track the changes made by the different users in the system, Who changed What and When, ValiProduction workflow will create a log file called **Security.log**.

The log file is saved in the subdirectory of the day like this one: ...\\MYAPPLIC\\RUNSAUTO\\YYYYMMDD_HHMMSS

For the day in reference, the log file (security.log) is unique and the successive changes are appended to the previous ones.

Example of security.log file:

```
[9/02/2012 09:46:04] INF 11085: App: D_000_MEB - Run: 20120116_000000 - User 'bruno' changes sta
[9/02/2012 10:40:47] INF 11086: App: D_000_MEB - Run: 20120116_000000 - User 'bruno' launch repr
[9/02/2012 10:43:00] INF 11085: App: D_000_MEB - Run: 20120116_000000 - User 'bruno' changes sta
[9/02/2012 10:46:27] INF 11085: App: D_000_MEB - Run: 20120116_000000 - User 'bruno' changes sta
[9/02/2012 10:46:37] INF 11085: App: D_000_MEB - Run: 20120116_000000 - User 'bruno' changes sta
```

MANUAL INPUTS

In the ValiProduction workflow, the User can change the Manual Inputs of the model via ValiForms.

A track changes log file is created in the directory of the ValiForms XML file, enabling to log all changes made by users.

The log file is created or updated when the ValiForms is “saved” or “saved and approved” and the name of the file is based on the ValiForms XML file (**XML_changes.log**).

Example of valiforms_changes.log file:

```
[12-03-13 17:34:09] INF 18033: User Christophe changed overwrite value of tag 111-FY-0035 from 9
[12-03-13 17:34:17] INF 18039: User Christophe changed product 111-NHT_STATUS from Shutdown to O
[12-03-13 17:34:20] INF 18041: User Christophe changed switch 111-NHT_REACTOR_DOWN from Running
[12-03-13 17:34:36] INF 18035: User Christophe changed final value of tank 111-NHT-HOLDUP_MASS1
[12-03-13 17:35:39] INF 18035: User Christophe changed final value of tank 111-NHT-HOLDUP_MASS2
```

SECTION 3: USING VALIMANAGER

3.1 STARTING VALIMANAGER

To start ValiManager:

- start ValiManager by selecting the ValiManager application in the Vali 4 Programs Start Menu
- log in using your user name and password. IP address of the LicenseServer can be changed using the drop-down list (see "[Belsim LicenseManager](#)" to add new IP addresses):



- if several applications are defined, ValiManager will present the list of ValiProduction applications to which you have access, otherwise the ValiProduction application is directly selected
- the list of automatic runs that are available is then displayed starting from the most recent ones

3.2 RUNS OVERVIEW

The "Overview" table is used to show to the user an overview of the whole process. By default the runs are displayed starting from the last run until the first day of the current month.

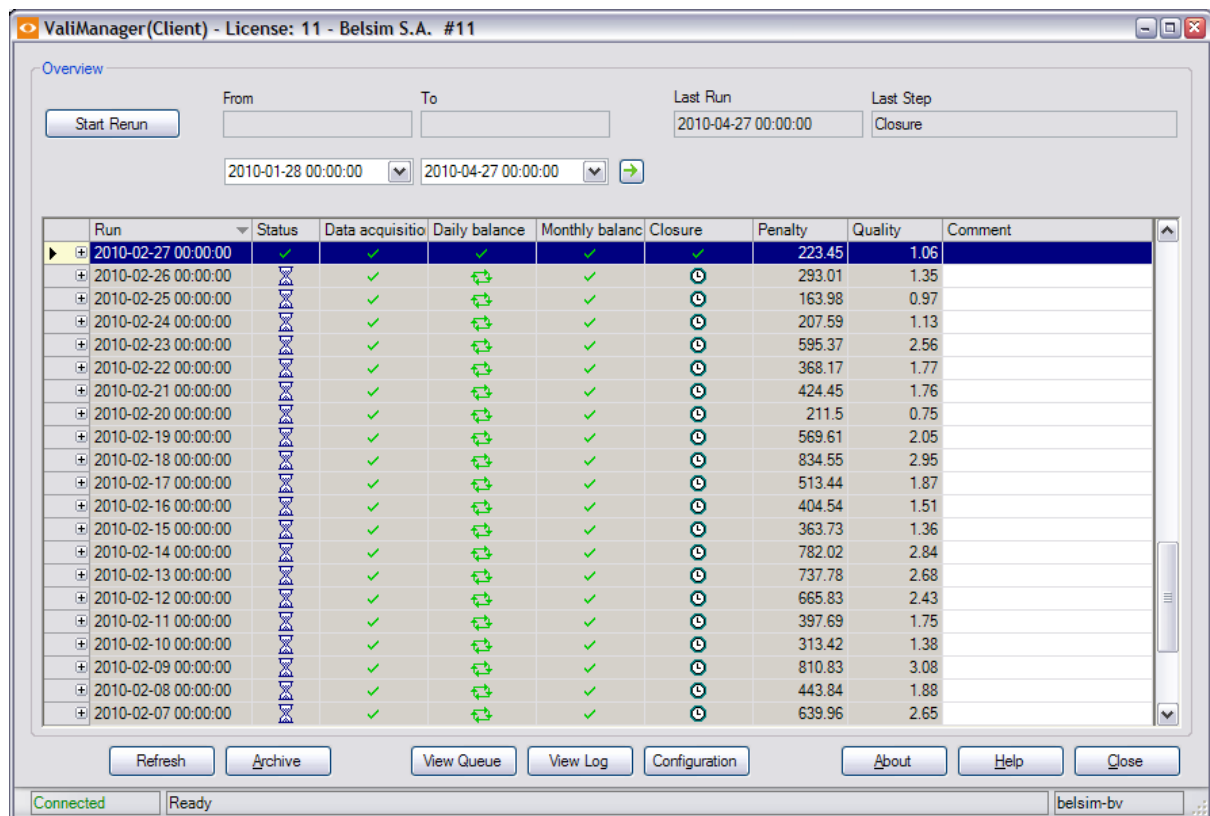


FIGURE 7 - VALIMANAGER - OVERVIEW OF THE RUNS

For rerun option, different information items are displayed:

- From: date of the first run to be reprocessed
- To: date of the last run to be reprocessed

In addition, two fields give information regarding the last run and the last step that have been processed:

- Last Run: date and time of the last processed run
- Last Step: Name of the last processed step

You can change the list of runs that are displayed by changing the start and end dates at the top of the page and pressing the  button.

The overview of the daily runs provides for each run the following information:

- Run: run date (starting date of the daily period)
- Status: global status of the run:
 -  : the run is still in progress
 -  : the run is approved
- Phases status: status of each of the phases of the workflow:
 -  : all steps of the phase are on hold
 -  : the steps are under progress
 -  : all steps have been completed
- Penalty: total penalty of the measurement corrections for the last validation of the run
- Quality: ratio of the Penalty by the Chi2 test value, a value above 1 indicating a suspicion of gross error
- Comment: comment on the run

Additional buttons are available:

- Refresh: manually refresh the graphical user interface. In any case, results are reloaded automatically every minute
- Archive: close and archive runs older than a date to be specified. After archiving, runs can not be reprocessed anymore

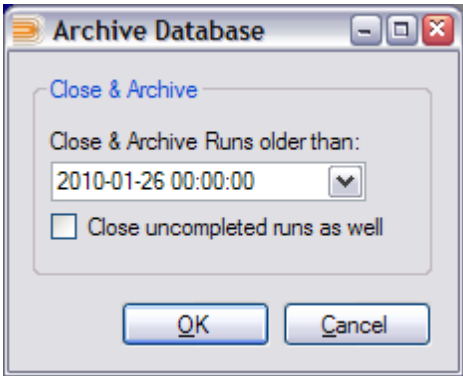


FIGURE 8 – VALIMANAGER - ARCHIVE PAGE

- View Queue: overview of the process queue list. This allows you:
 - to see if other applications are running
 - to determine how many runs are waiting in the process queue list

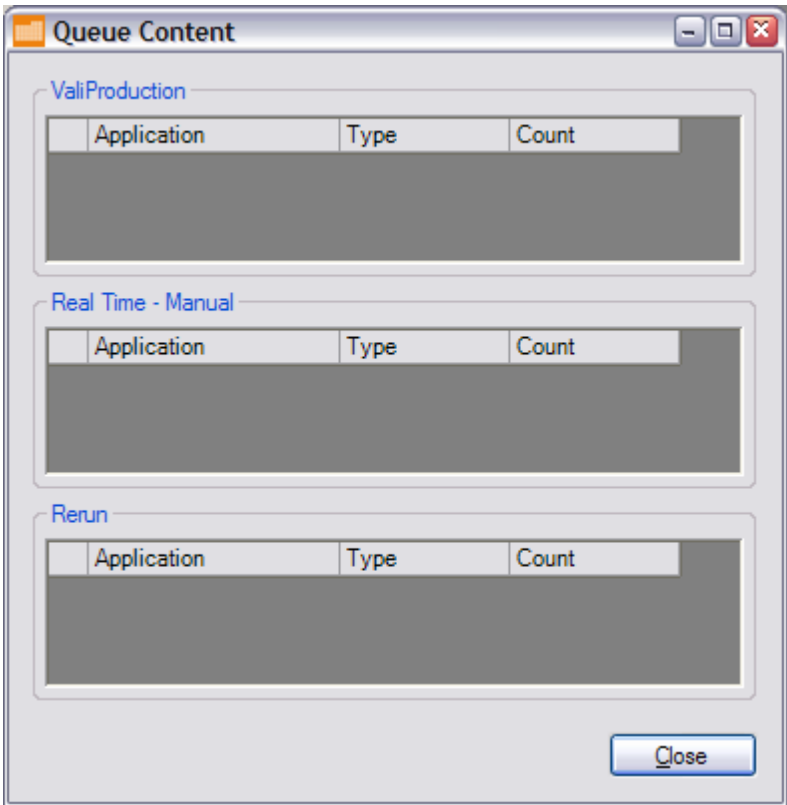


FIGURE 9 – VALIMANAGER - VIEW QUEUE PAGE

- View Log: display the content of the ValiService log file located on the remote server

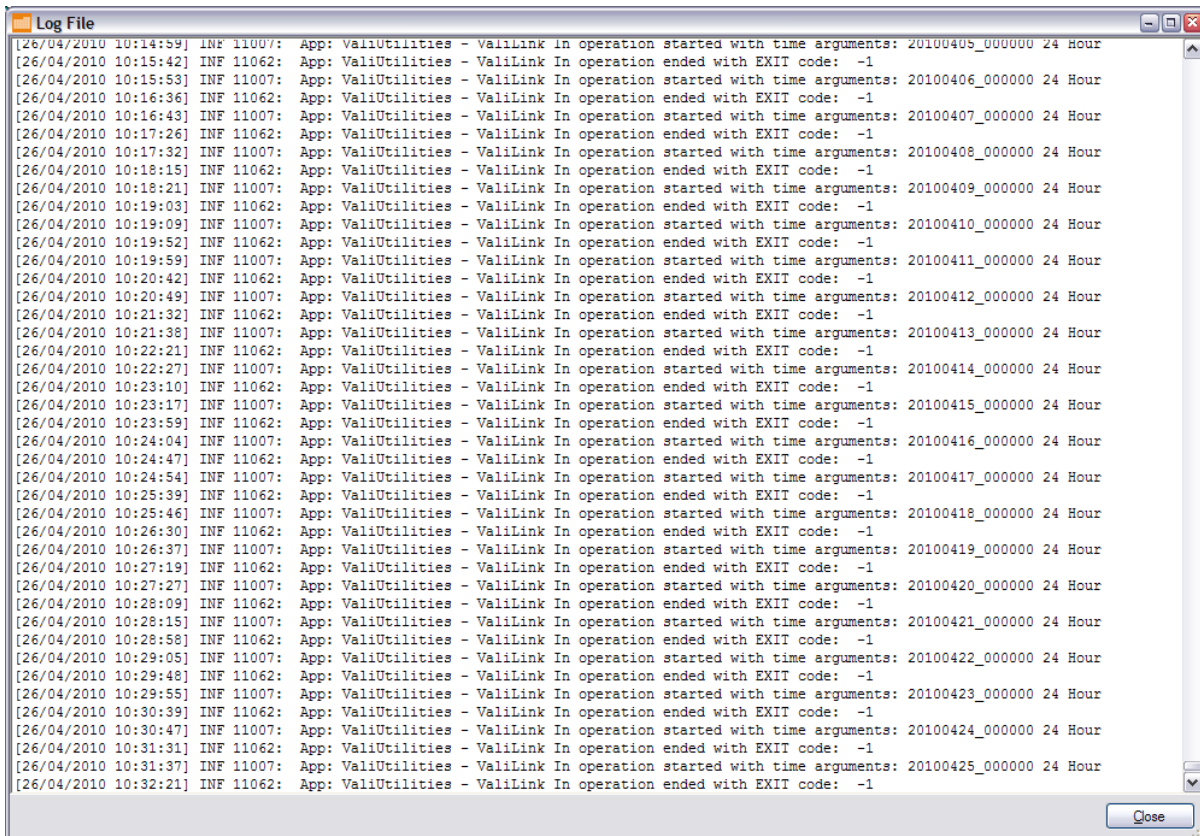


FIGURE 10 – VALIMANAGER - VIEW LOG PAGE

- Configuration: opens the configuration dialog box.
 - The **General** tab allows you to define the schedule settings:
 - Data Time: beginning time of the validation periods
 - Process Time: time of the day when the new runs are created
 - Timer Tick: interval to check if an action has to be performed by ValiService
 - Vali Time Out: time period allowed for the validation run. If the run lasts for a longer time period, the run is cancelled. No time out will be applied if this value is negative

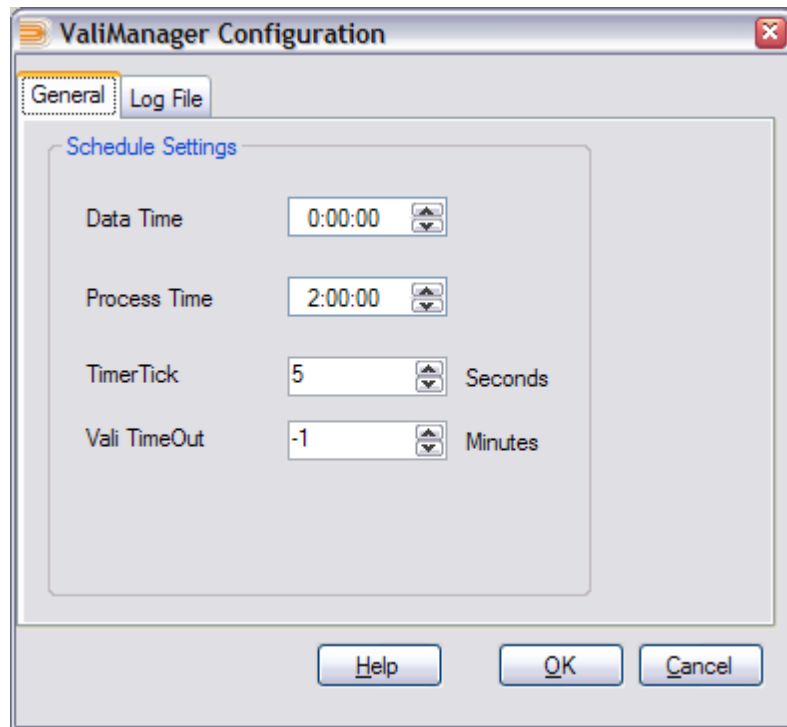


FIGURE 11 – VALIMANAGER – SCHEDULE OF AUTOMATIC RUNS

- The **Log File** tab allows you to define properties of ValiService Log file:
 - Path: path where the Log File will be generated by ValiService (by default [InstallDir]\belsim\App)
 - File Name: name of the log file
 - Error level to log: level of errors to log
 - Backup: option: type of backup used
 - Maximum Size: maximum size of the file

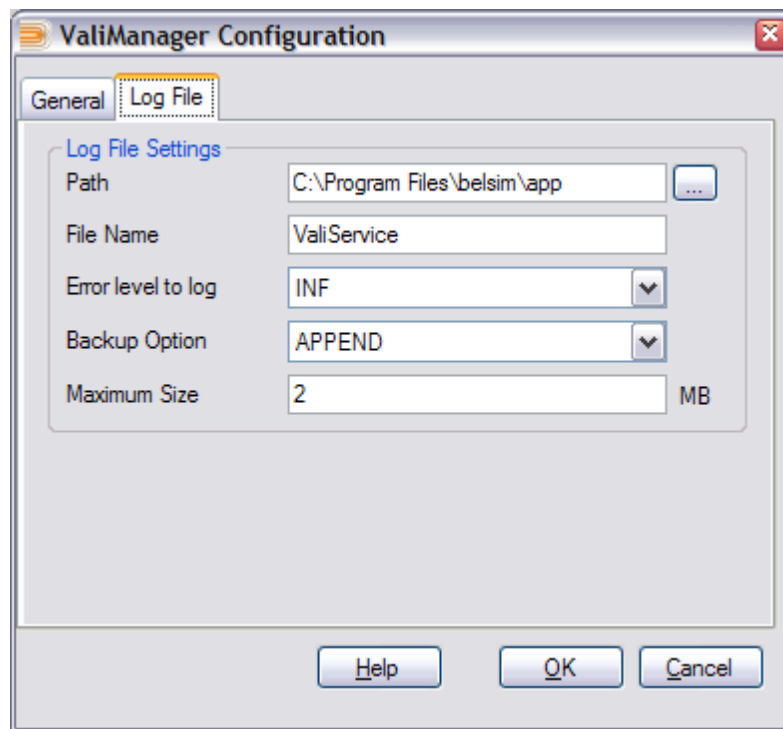


FIGURE 12 – VALIMANAGER- LOG FILE OF VALISERVICE

These settings are saved in the ValiProduction.ini file located in the App folder of VALI installation.

The use of the different buttons of the ValiManager is restricted depending on user roles:

BUTTON	USER ROLES					CORRESPONDING ACTION
	MASTE R	OPERAT OR	CONFIGURA TOR	MODEL LER	READER	
Refresh	x	x	x	x	x	Refreshes the list of runs
Archive	x					Closes and archives runs
View Queue	x	x	x	x	x	Displays an overview of the process queue list
View Log	x	x	x	x	x	Displays the content of the ValiService log file located on the remote server
Configurat ion	x					Displays the configuration settings
Start Rerun	x					Reprocesses runs

TABLE 6 - USER ROLES

3.3 DIGGING INTO RUNS

Clicking on the  button at the left of the run date expands the information on the run.

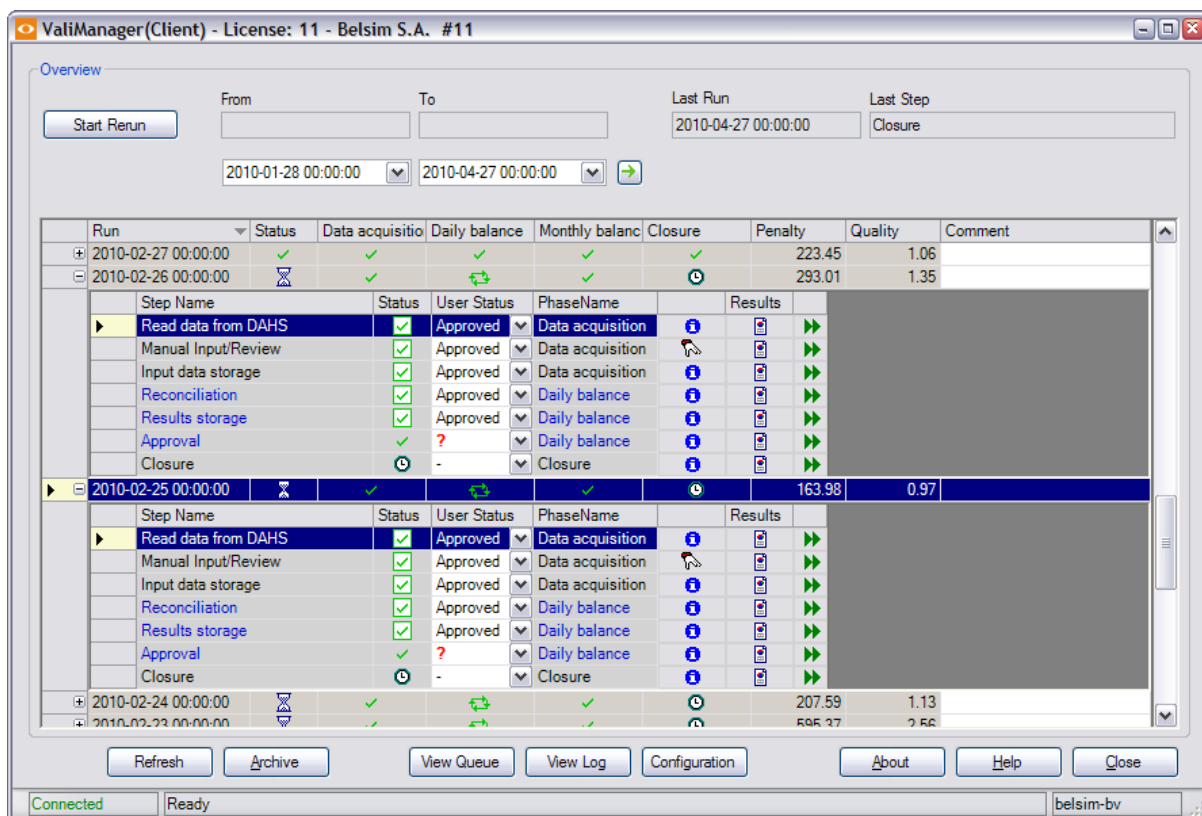


FIGURE 13 – VALIMANAGER - EXPANDED VIEW OF THE RUNS

For each step, the following information is given:

- Step Name: short description of the step. Moving the mouse above the short description displays the long description of the step
- Status: status of the step:
 -  : ON HOLD: the conditions of the steps are not yet satisfied
 -  : PENDING: the conditions are now satisfied; the action is pending for user input
 -  : QUEUED: the conditions are now satisfied, the action has been queued
 -  : IN PROGRESS: the action is currently executed
 -  : DONE: the step has been successfully executed
 -  : WARNING: the step has been executed and reported a warning status
 -  : FAILED: the execution failed
 -  : APPROVED: the step has been either automatically or manually approved

-  : SKIPPED: the user decided to skip the step (if allowed by the configuration)
- User status: action of the user. The field can only be modified by authorized users. Depending on the task status, the user can:
 - Rerun the task, after some corrections if required
 - Approve the results, setting the step APPROVED
 - Skip the task, setting the step SKIPPED
- PhaseName: name of the phase to which the step belongs
-  /  : button to open the log file of the task/button to start ValiForms for manual input from the user
-  : button to access the results of the task
-  : button to reprocess the step. Reprocessing a step will reset to ON HOLD all the steps that are depending on the current step in the logical execution sequence, according to the step conditions (in the current run as well as in the following runs)

Right click on the date of a run allows to open the corresponding run directory (where the application files are located) of the **RunsAuto** folder. In addition to the application files (model file, measurement files, reconciliation results, initialization files,...), each run directory contains a log file ("security.log") to track activities performed by the different users through ValiManager on a run by run basis.

Example:

```
[9/02/2012 09:46:04] INF 11085: App: MYAPPLIC - Run: 20120116_000000 - User 'Bruno' changes statu
[9/02/2012 10:40:47] INF 11086: App: MYAPPLIC - Run: 20120116_000000 - User 'Bruno' launch repro
```

3.4 STARTING RERUN

To start the execution of rerun tasks, click on "Start Rerun" button. A configuration window will be opened with the runs settings. The "Start Rerun" page is used to specify the date of the first run and the first step that will be reprocessed.

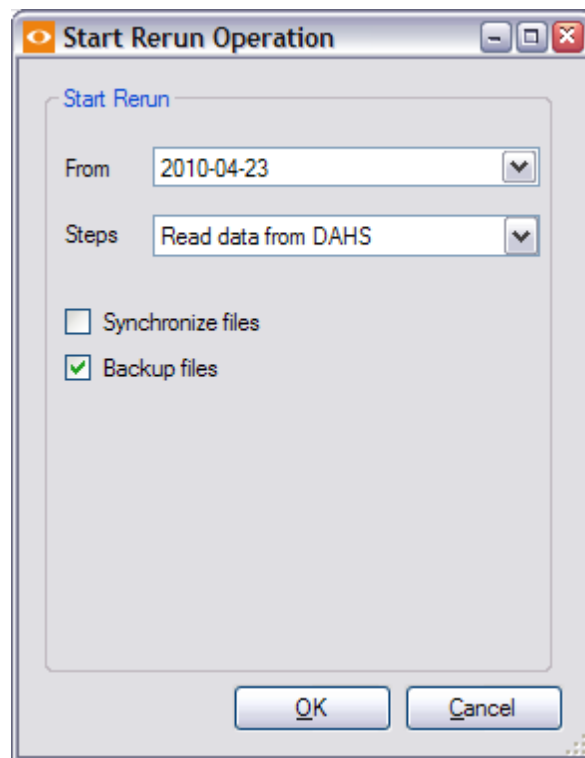


FIGURE 14 – VALIMANAGER-START RERUN

- From: date of the first run to be reprocessed. The rerun will affect all the runs in between this starting date and the date of the last created run
- Steps: step to be reprocessed. Reprocessing a step will reset to ON HOLD all the steps that are depending on the selected step in the logical execution sequence, according to the step conditions (in the current run as well as in the following runs) except the steps that have been SKIPPED
- Synchronize files: if selected, the list of files available in the official directory of your ValiManager application appears on the left hand side of the window. Selected files will first be copied from the official directory to the directories of all runs to be reprocessed
- Backup: if selected, ValiManager will save the directories of all runs to be reprocessed

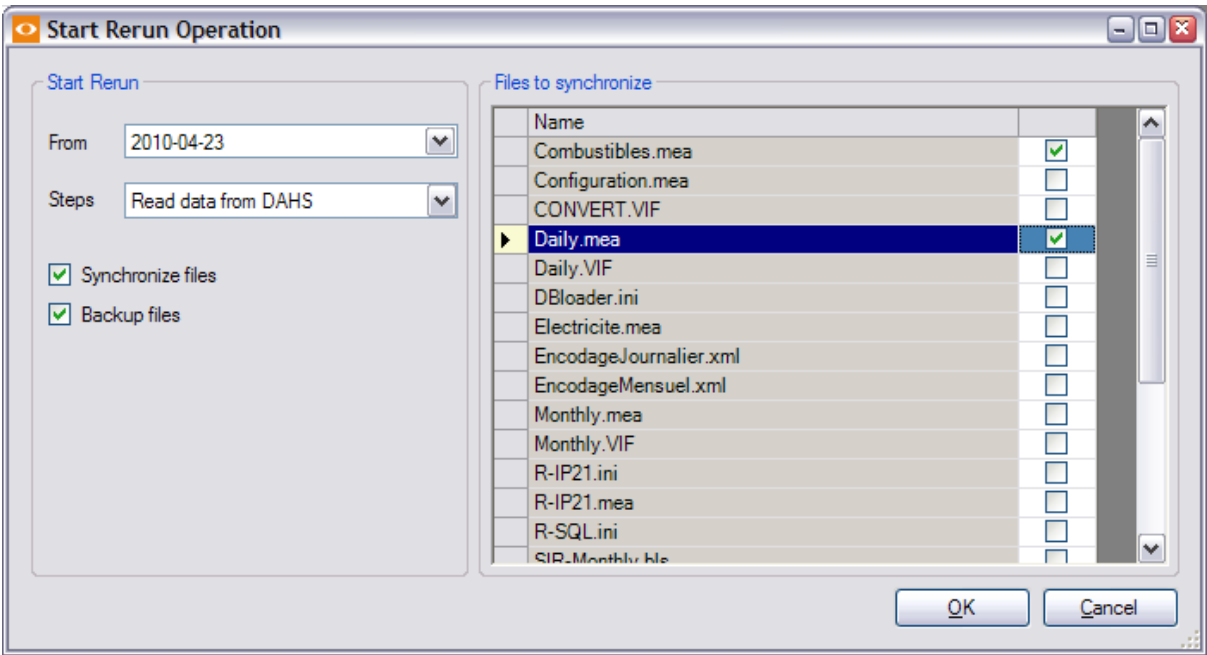


FIGURE 15 – VALIMANAGER- START RERUN AND SYNCHRONIZE FILES



belsim

CHAPTER 17

VALIFORMS

SECTION 1: INTRODUCTION

1.1 VALIFORMS OBJECTIVES

ValiForms is a configurable user interface primarily used to handle manual inputs of a Vali application through multi-pages forms:

- entering manual input numeric data
- entering corrections on automatically acquired data
- associating products to storage tanks and streams
- entering model settings like modes of operation

The manual inputs will be stored in one or several mea file, to be later read by ValiEngine or ValiDBLoader.

ValiForms is also used to generate pre-configured reports based on the data stored in ValiDatabase:

- data extracted from a single run
- aggregated data from several runs
- graphical trends over several runs

The reports can either be browsed in interactive mode (and possibly printed as PDF file) or directly generated as PDF files, in background mode.

The Forms are configured using [ValiFormsBuilder](#) application.

VALIFORMS USED FOR INPUT

When used as Input user interface, ValiForms uses at minimum two files:

- an xml file, generated with ValiFormsBuilder application, describing the contents of the Forms
- a mea file, generated by ValiForms with the user input data

Optionally, ValiForms can use additional information sources:

- an auxiliary mea file, giving the initial counter and tank inventory data
- an historian file with measured values coming from an external historian (typically generated by ValiLink)
- a ValiDatabase, giving validated values and initial counter and tank inventory data

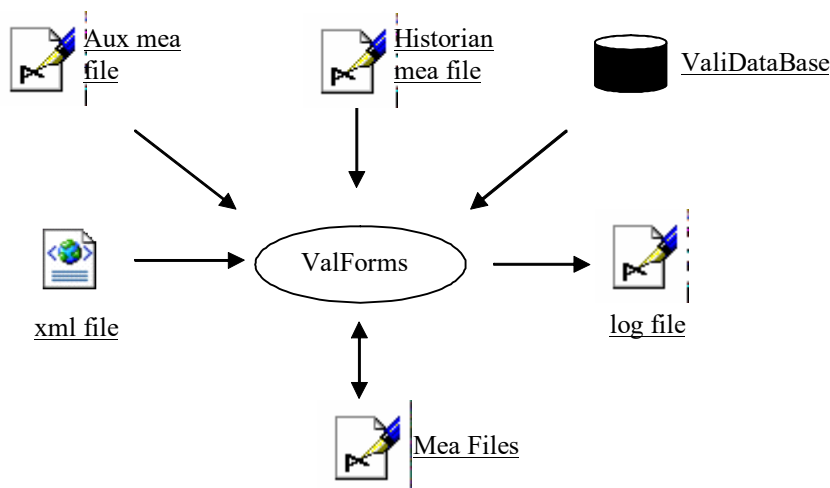


FIGURE 1 - VALIFORMS INPUT ENVIRONMENT

ValiForms can handle several mea files, each mea file being displayed in a separate section (Area 1, Area 2, etc... in the example below).

Each section is split up in several sub-sections (Unit 1, Raw Materials, Unit 2/3 in the example below), enabling to logically group and organize related input data.

Each sub-section contains a number of fields defining input data (tags, tanks, counters, switches, etc.) which will generate one or several lines in the mea files.

Description	Tag Name	Initial	Final	Corrected	Validated	Delta	Delta Val	Eng Units	Imbalance
Storage T1	1LMB4271_BI.TJ	0						TON	☐
Storage T2	1LMB4352_BI.TJ	0						TON	☐
Storage T3	1LMB4702_BI.TJ	0						TON	☐
Storage T4	7LMB3011_BI.TJ	0						TON	☐
Storage T5	9LMB465_BI.TJ	0						TON	☐
Storage T6	9LMB466_BI.TJ	0						TON	☐

Description	Tag Name	Measured	Corrected	Validated	Eng Units
Third party	9FMB8011_BI.TJ	-9999			TON/D
Product 1 97% -> T4	1FMB4721.TJ_1	-9999			M3/D
Product 1 95% -> T5	7FMB3031.TJ_1	-9999			M3/D

Description	Tag Name	Measured	Corrected	Validated	Eng Units
Product 1 97% -> Unit 2	1FMB1211.TJ_1	-9999			TON/D
Product 1 95% -> Unit 2	1FMB1213.TJ_1	-9999			TON/D
Product 1 97% -> Unit 3	1FMB2211.TJ_1	-9999			TON/D
Product 1 95% -> Unit 3	1FMB2213.TJ_1	-9999			TON/D
Product 1 -> Unit 2/3	SWT-NAM				TON/D
	1FMB1215.TJ_1	-9999			TON/D
Loading 93%	NAM-C93N_MAS	0			TON/D
Condensat	1FMB4603.TJ_1	-9999			TON/D

Description	Tag Name	Measured	Corrected	Validated	Eng Units
Dilution 97 -> 95%	1FMB4722.TJ_1	-9999			M3/D
Dilution 95 -> 93%	7FMB3111.TJ_1	-9999			M3/D

FIGURE 2 - A TYPICAL VALIFORMS INPUT FORM

VALIFORMS USED FOR OUTPUT

When used as Output user interface, ValiForms uses:

- an xml file, generated with ValiFormsBuilder application, describing the contents of the Forms
- ValiDatabase, from which the output data are extracted

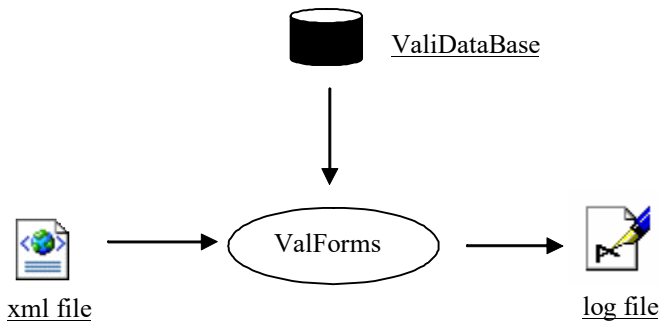


FIGURE 3 - VALIFORMS OUTPUT ENVIRONMENT

As for input Forms, the report can be split up in several sections (Rapports in the example below)

and sub-sections (Unit1 and Unit 2/3, etc... in the example below).

Each sub-section contains a number of fields defining the report (tag values, tank data, trends, etc.).

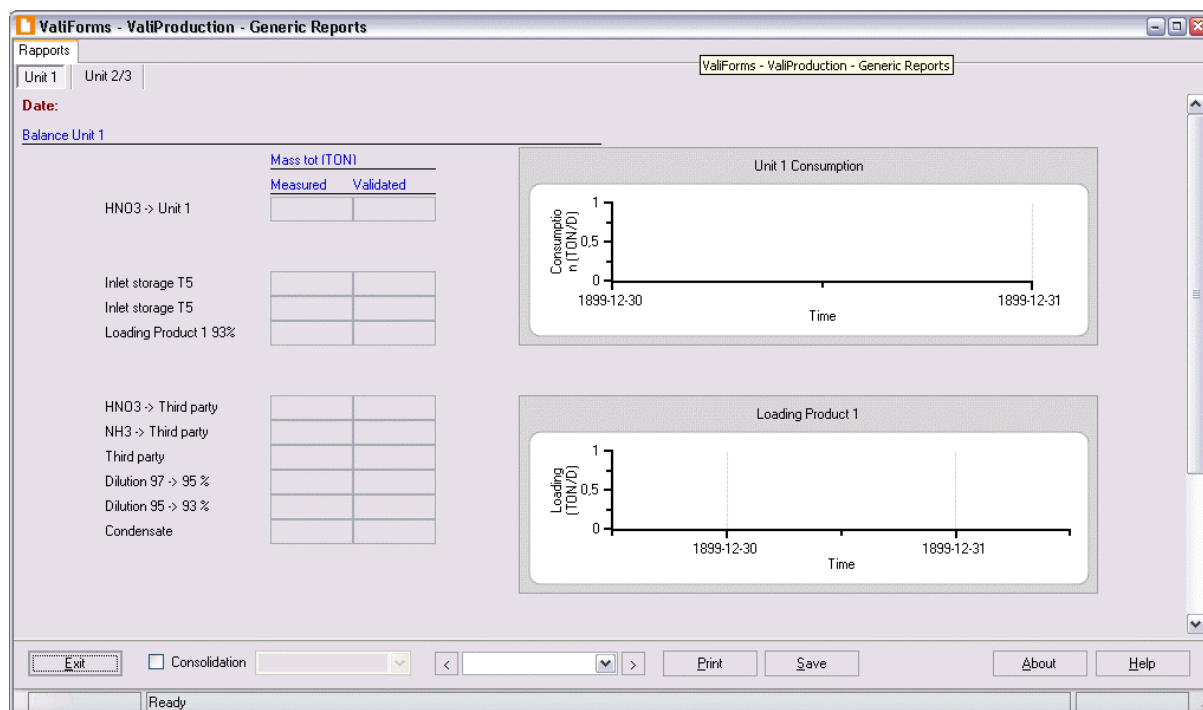


FIGURE 4 - A TYPICAL VALIFORMS OUTPUT FORM

1.2 USING VALIFORMS IN A VALIPRODUCTION APPLICATION

When used in a ValiProduction application, ValiForms can be called from ValiManager, presenting the data related to the currently selected run and step. The user can browse through the data from different daily runs, but only the data relative to the current run and current section can be edited by the user.

When saving the data, the user can also approve the step, indicating that the input data have been entered and checked, so that the current step can be considered as completed.

1.3 USING VALIFORMS IN A VALIPERFORMANCE APPLICATION

ValiForms can be called by ValiScheduler/ValiService to automatically generate PDF reports during the storage phase of a run. The Path field of ValiScheduler enable to specify the name of the ValiForms xml file. Optionally, you can also indicate whether the input or output sections must be printed (options -I and -O) and specify the aggregation period (options -A).

More information is provided on ValiForms arguments in the section [Starting ValiForms](#).

SECTION 2: CONFIGURING VALIFORMS

2.1 USING VALIFORMSBUILDER

ValiFormsBuilder is started by double-clicking the ValiFormsBuilder.exe file in bin directory of VALI installation folder.

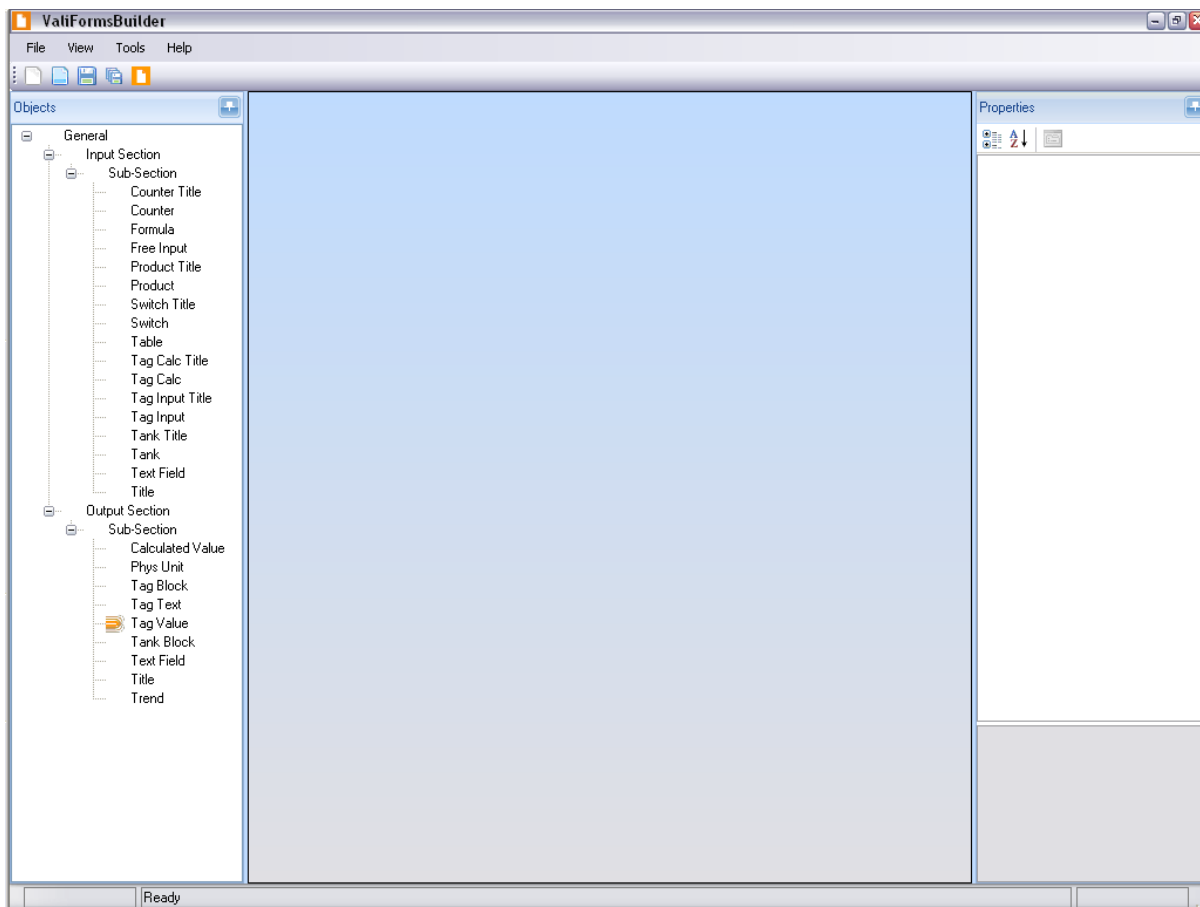


FIGURE 5 - EMPTY VALIFORMSBUILDER WINDOW

Main window is divided in 6 main parts:

- The menu at the top

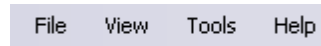


FIGURE 6 - VALIFORMSBUILDER MENU

- The toolbar, under the menu bar



FIGURE 7 - VALIFORMSBUILDER TOOLBAR

- The objects panel, containing the tree view, located on the left

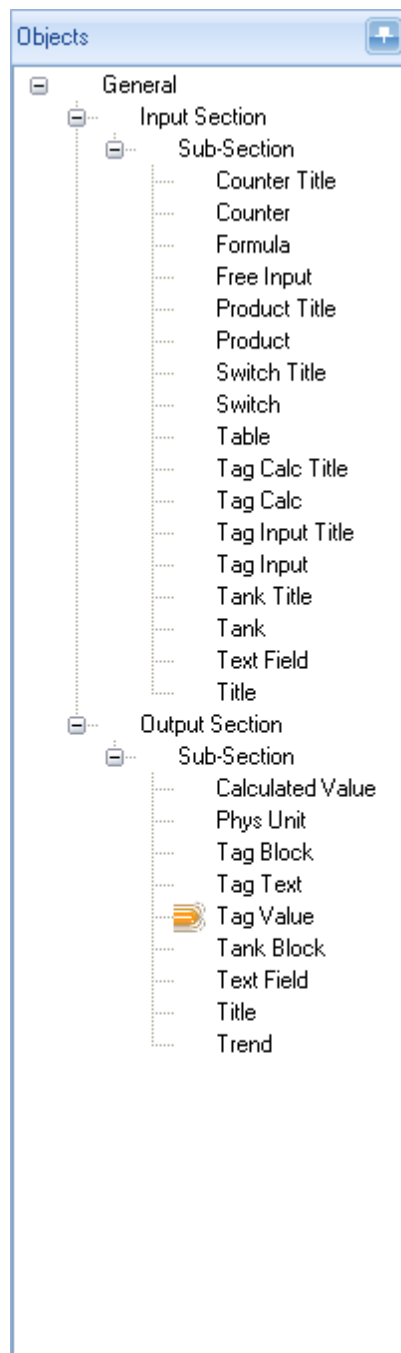


FIGURE 8 - VALIFORMSBUILDER TREE VIEW

- The tabs/sheet view, at the center of the window

ValiFormsBuilder.xml

Input2 Input 1 Output full Output 2 Monthly input

SubSection

Description	Tag Name	Initial	Final	Converted	Overwrite	Validated	Phys. L
Counter example	P14-P141-B1185						kW

Description	Tag Name	product
tank field	31F01	LPG

Description	Tag Name	Mode
		Option 1

	Column1	Column2	Total
► 50FI0070	52		
50FI0072			
Total			

Description	Tag Name	Value	Converted	Overwrite	Validated	Phys. Unit
Comment	54FI1012					t/h

Description	Tag Name	Value	Overwrite	Validated	Phys. Unit
	54FI1012				

Description	Tag Name	Initial	Final	Overwrite	Validated	Delta	Delta V

Calculations

Avg of 50FI0070 and 50FI0072 (measured) t/h

Avg of 50FI0070 and 50FI0072 (validated) t/h

Description	Tag Name	Value	Overwrite	Validated	Phys. Unit
	50FI0070				

FIGURE 9 - VALIFORMSBUILDER TABS VIEW

- The properties panel, located on the right

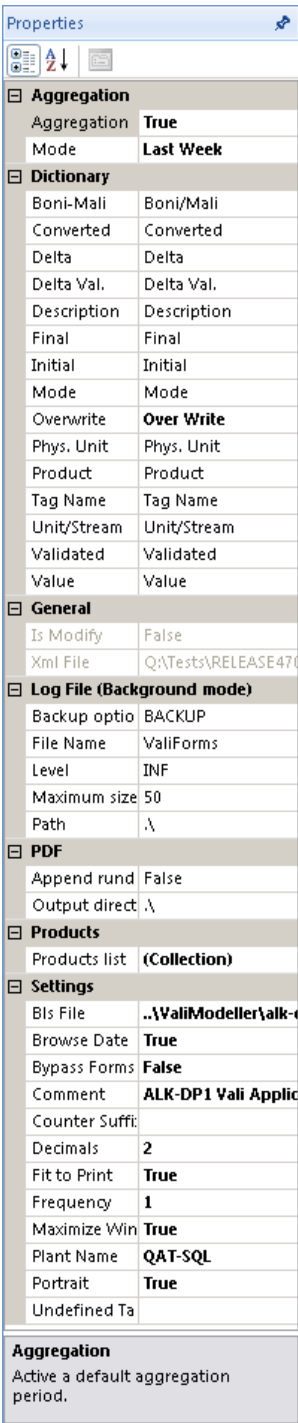


FIGURE 10 - VALIFORMSBUILDER PROPERTIES PANEL

- The status bar, located at the bottom



FIGURE 11 - VALIFORMSBUILDER STATUS BAR

MENU BAR

The menu is located at the top of the window . It provides a quick access to all main functionalities of the program:

- File
 -  New: creates a new file (keyboard shortcut: ctrl+n; drag and drop the General node of the tree on the left)
 -  Open: open an existing file
 -  Save: saves current file (keyboard shortcut: ctrl+s)
 -  Save All: saves all files opened in current session of the program (keyboard shortcut: ctrl+shift+n)
 - Save As: saves current file under a specified name
 - Exit: closes program (keyboard shortcut: alt+F4). If some files are unsaved, a prompt invites you to save modified files.

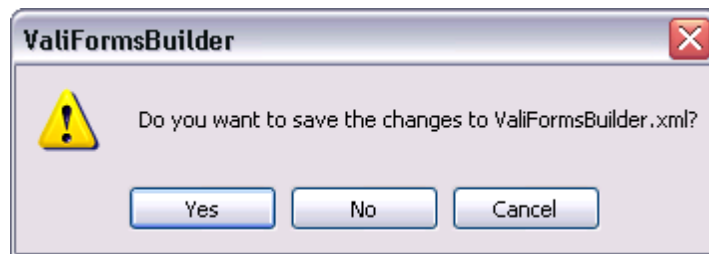


FIGURE 12 - SAVE MODIFICATION WINDOW

- View
 - Full Screen: toggle view to full screen mode (removes window header)/normal view (keyboard shortcut: F11)
 - Grid: Shows/hide the grid (keyboard shortcut: ctrl+g)
 - Fields:
 - Background: grays background of fields
 - Border: shows/hide a border around fields
- Tools
 -  ValiForms: starts ValiForms application and opens current file
- Help
 - Manual: opens ValiFormsBuilder help
 - About: displays general information about the application

TOOLBAR

Toolbar is located below menu. It provides quick access to most common functions. A description of the function is given if the mouse remains on a button.

-  create a new file
-  open an existing file
-  save current file
-  save all open files
-  starts ValiForms application and opens current file

OBJECTS PANEL

The objects panel contains the object tree view. All objects that can be added on a ValiForms are contained in this tree. To add an object, drag it from the tree and drop it on the tab view. The different types of objects are described in [input section fields](#) and [output section fields](#).

The objects panel can be hidden using the pin located at the top-right of the panel. The panel will then auto-show when the mouse is placed on the tab located on the left of the window.

TABS VIEW

The tabs view is the most important part of the window. This is the area where sections, sub-sections and fields are placed.

PROPERTIES PANEL

The properties panel is the place where properties of currently selected object are displayed. Properties can be changed on this panel.

The properties panel can be hidden using the pin located at the top-right of the panel. The panel will then auto-show when the mouse is placed on the tab located on the right of the window.

STATUS BAR

The status bar displays general information about currently loaded file.

2.1.1 DEFINING VALIFORMS MAIN PROPERTIES

The different properties of current ValiForms file are available on the right-hand part of the window, when the file tab is selected.

AGGREGATION

The aggregation settings define a default aggregation period for the Form:

- Aggregation: use aggregation
- Mode: mode of aggregation:
 - Month to Date
 - Last Month
 - Last Week
 - Week to Date
 - Last Year
 - Year to date
- Only converged Runs: average aggregation will be carried out only on converged runs. Results of non-converged runs will be excluded from the calculation. This parameter only affects average aggregation
- Only converged Runs: average aggregation will be carried out only when tags are actually used in the model (InUse=1). This parameter only affects average aggregation

[-] Aggregation	
Aggregation	False
Mode	
Only converged	False
Only tags when	False
[+] Dictionary	
[+] General	
[+] Log File (Background mode)	
[+] PDF	
[+] Products	
[+] Settings	

FIGURE 13 - AGGREGATION MODE

DICTIONARY

The dictionary shows the automatic title strings that will be written when generating titles. You can choose your own naming convention for the listed fields.

	Aggregation	
	Dictionary	
	Boni-Mali	Boni/Mali
	Converted	Converted
	Delta	Delta
	Delta Val.	Delta Val.
	Description	Description
	Final	Final
	Initial	Initial
	Mode	Mode
	Overwrite	Overwrite
	Phys. Unit	Phys. Unit
	Product	Product
	Tag Name	Tag Name
	Unit/Stream	Unit/Stream
	Validated	Validated
	Value	Value
	General	
	PDF	
	Products	
	Settings	

FIGURE 14 - VALIFORMS DICTIONARY

LOG FILE

	Aggregation	
	Dictionary	
	General	
	Log File (Background mode)	
	Backup optio	BACKUP
	File Name	ValiForms
	Level	INF
	Maximum size	50
	Path	.\
	PDF	
	Products	
	Settings	

FIGURE 15 - VALIFORMS LOG FILE

The log file settings are only used when ValiForms is called in background mode. This section is not mandatory; default data will be used in place of any missing information item. Following fields have to be provided:

- Back-up option:
 - BACKUP: the old log is renamed by appending _log to the filename and the messages of the current run are logged into a new file (default setting)
 - ARCHIVE: messages are written in a new file named by appending the current run date to the selected file name
 - APPEND: messages are written at the end of the existing log file
- File Name: name of the log file (the extension .log will be automatically added)
- Level: used to filter the level of information and error messages to be written in the log file. The log level can be one of the following: INF, WRN, ERR, ERS, ERF. Only messages with levels higher than the log level are written. For instance, with ERR specified as log level, only ERR, ERS and ERF message will be printed
- Size: the maximum size of the log file expressed in Mb
- Path: path of the log file (by default, path of the xml file)

PDF

PDF options allow you to specify options when generating PDF files using ValiForms in background mode:

- Output directory: the directory where the output file will be placed
- With rundate: rundate will be appended to the PDF file name

☒ Aggregation	
☒ Dictionary	
☒ General	
☒ PDF	
Output director	.\
With rundate	False
☒ Products	
☒ Settings	

FIGURE 16 - PDF

PRODUCTS LISTS

Products allows you to define list of products that will be used in [products fields](#).

- Products list: define the list of products

☒ Aggregation	
☒ Dictionary	
☒ General	
☒ PDF	
☒ Products	
Products list	(Collection)
☒ Settings	

FIGURE 17 - PRODUCTS

To edit the products list, click on the  button that appears when clicking on (Collection). This button will open a new window that allows you to create new products list:

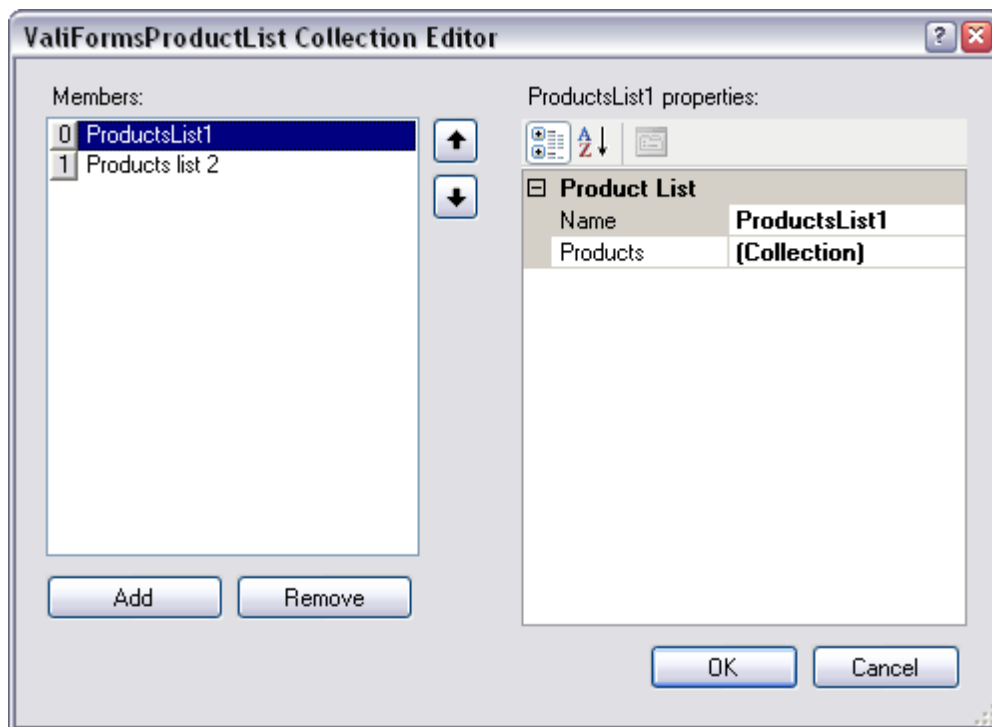


FIGURE 18 - PRODUCTS LIST EDITOR

To add a new products list, click on the Add button. To remove a list, click on the Remove button. The Name of the collection is defined in the right part of the window. To define the list of Products in a list, click on the  button that appears when clicking on (Collection). A new window appears that allows you to define the list of products (one per line):

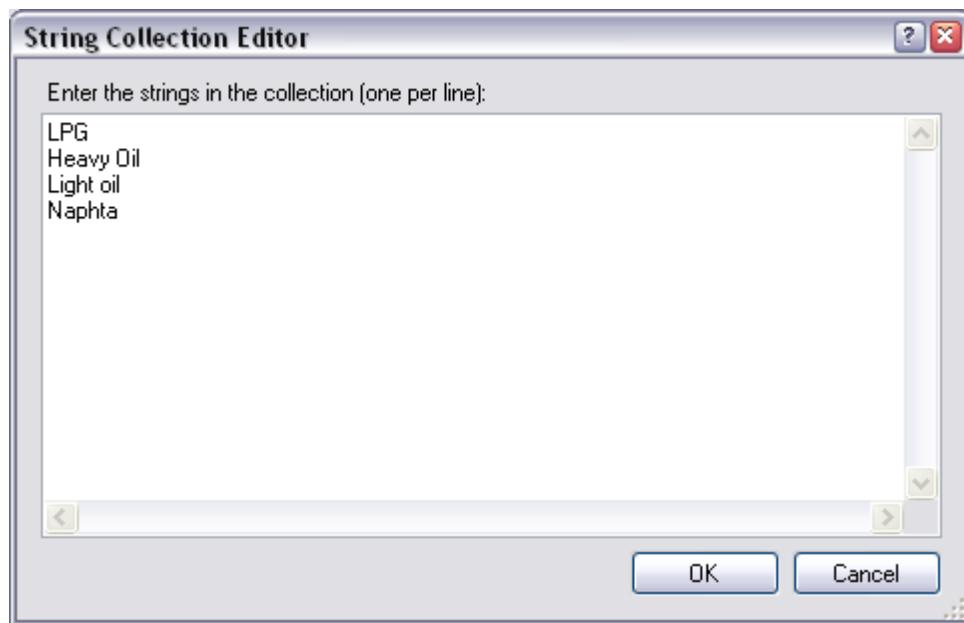


FIGURE 19 - LIST OF PRODUCTS IN A PRODUCTS LIST

SETTINGS

Setting allows to define some other general settings linked to the configuration of ValiForms.

A has the following properties:

- Bls File: associated bls model file
- Browse Date: a date field will be displayed at the bottom of the Form, enabling to change the run date
- Bypass Forms for undefined tags: when checked, the Undefined Tags Mea file will be created.
- Comment: displayed in the window title bar of ValiForms
- Counter Suffix: suffix added to counter names when converting counter values to rates
- Decimals: default number of decimals for numeric values
- Fit to print: the print-out will be scaled to fit the printing area
- Frequency: frequency of the runs expressed in days. Use to get initial counter and tank inventories in daily mode
- Maximize Window: ValiForms window will be maximized at application start-up
- Plant name: associated plant name (defining ValiDatabase)
- Portrait: the form will be printed in portrait mode (default is landscape)
- Products Dictionary: path of Products dictionary file
- Track Changes: will create a _changes.log file in the directory of the ValiForms XML file, enabling to log all changes made by users.
- Undefined Tags Mea: name of the file that will be written with all tags that are in the historian but not referred to in ValiForms

+ Aggregation	
+ Dictionary	
+ General	
+ Log File (Background mode)	
+ PDF	
+ Products	
- Settings	
Bls File	SIR.bls
Browse Date	False
Bypass Forms f	False
Comment	ValiForms
Counter Suffix	.cv
Decimals	2
Fit to Print	False
Frequency	1
Maximize Wind	False
Plant Name	SIR-SQL
Portrait	False
Products Dictio	dictionary.mea
Track Changes	False
Undefined Tag	Remaining.mea

FIGURE 20 - VALIFORMS SETTINGS

ADDING AND DELETING

ValiFormsBuilder is all about drag & drop. To add any kind of objects (open or create file, add a section, sub-section or field), simply drag it from the tree view and drop it onto the file.

To delete an object, right-click on it and select Delete. Field can also be deleted by right-clicking and selecting delete.

2.1.2 CONFIGURING AN INPUT SECTION

Input sections enables creating forms where values can be manually input or modified.

To create an input section, drag the Input section node from the tree and drop it on the working sheet. This will create a new section tab. Input sections are divided in sub-sections where input fields are placed. An input sub-section is created by dragging the input sub-section from the tree view.

To create an input form, it is required to have at least one input section containing at least one input sub-section. A field is added by dragging it from the tree view and dropping it on the sub-section sheet.

Input sub-section is a grid containing two columns and an unlimited number of lines. A field can be placed on any free space of the grid. The grid can be showed/hidden using keyboard shortcut ctrl+g or by selecting **View|Grid**.

The different fields that are available are

- Counter Title: line describing the values of a counter field
- Counter: data of a counter, from which a rate will be computed

- Formula: tag value calculated from other input data
- Free Input: freely formatted input data
- Product: definition of the product associated to a stream or a unit
- ProductTitle: line describing the values of a product field
- Switch: definition of a model switch
- SwitchTitle: line describing the values of a switch field
- Table: definition of a group of input data displayed as a table
- TagCalc: calculated tag based on a given value times a predefined coefficient
- TagCalcTitle: line describing the values of a TagCalc field
- TagInput: tag value
- TagInputTitle: line describing the values of a TagInput field
- Tank: data associated to a storage Tank
- TankTitle: line describing the values of a Tank field
- Text Field: text to be displayed in the Form
- Title: title text (underlined) to be displayed in the Form

Some fields have a counterpart named FieldNameTitle, that simply represents titles to be displayed above corresponding field, to give more information above the different parts of a field

PROPERTIES OF AN INPUT SECTION

An input section allows to define input sub-sections.

An input section has the following properties:

- **Aggreg..mode:** default aggregation mode of the section, used as default when creating new sub-sections
- **Approval:** data must be approved
- **Historian File:** name of the mea file extracted form the historian and used as primary input to create the mea file of the section
- **Init From DB:** take initial value of counters and tank inventories from ValiDatabase rather than from the auxiliary mea file (Mea aux)
- **Mea aux:** auxiliary mea file giving the initial value of counters and tank inventories
- **Mea File:** name of the mea file created by the section
- **Name:** section name displayed in the main tabs of the Form
- **Overwrite with historian:** measurements as available in the main Mea file is always overwritten by the data from the historian file

General	
Aggreg. Mode	VALUE_BEGINDA
Approval	False
Historian File	R-IP21.mea
Init From DB	1
Mea Aux	
Mea File	Input2.mea
Name	Input2
Overwrite With	False
Style	Input

FIGURE 21 - INPUT SECTION PROPERTIES

PROPERTIES OF AN INPUT SUB-SECTION

An input sub-section allows to define input fields.

An input sub-section has the following properties:

- **Aggreg.Mode:** default aggregation mode of the sub-section, used as default when creating new fields
- **Initial Value:** the initial value of counters and tank inventories must be taken from the preceding day or month
- **Name:** sub-section name displayed in the secondary tabs of the section
- **Show Date:** run date must be shown at the top of the sub-section. When aggregating data, the begin and end dates of the aggregation period will be displayed
- **Show Quality:** run quality indicator must be displayed at the top of the sub-section (Red: quality > 2.5, Green: quality < 1.5, orange otherwise). When aggregating data, the quality indicator is not shown

General	
Aggreg. Mode	
Initial Value	DAILY
Name	SubSection
Show Date	False
Show Quality	False

FIGURE 22 - INPUT SUB-SECTION PROPERTIES

COMMON PROPERTIES

Following properties are common to all fields:

- **Comment:** some text that is displayed in front of the control (not used for title fields)
- **Name:** name of the control, also used as name of the tag associated with the control (tag for which value will be retrieved, or tag for which output will be written). This property is mandatory for Free Input field
- **Location:** the location of the control on the grid, indicated by the row and column number

All fields can be duplicated. To duplicate a field, right click it and select Duplicate.

THE DIFFERENT FIELDS

Counter

A counter represents data of a counter, from which a rate will be computed

Counter measurements are converted to an actual measured value using the difference between the final and initial values of the counter and a conversion coefficient:

Converted = (Final - Initial) * Coefficient

Depending on the section settings (**Init from DB** and **Mea Aux**), the initial value will either be read from ValiDatabase or from the auxiliary mea file.

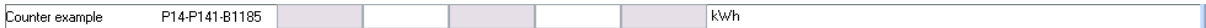


FIGURE 23 - COUNTER FIELD

A counter field has the following properties:

- Coefficient: coefficient to convert the counter value to a rate
- Counter Max: the maximum value the counter can reach before resetting to zero
- Default Value: the default value of the counter; =BLS to read default value from bls model file
- Init Value: an initial value will be read from the aux. mea file or from ValiDabatase. Otherwise, the initial value is set to 0
- Mandatory: a valid value must be provided to Approve the data
- OverWrite: the converted value can be overwritten by the user
- Phys. Units: the physical units of the rate (after conversion from the counter); =BLS to read default value from bls model file
- Tag Name: name of the converted tag. If left blank the name of the tag will be built from the counter name and the counter suffix (ValiForms setting)
- Validated: the validated value will be shown
- Write Value: the Final counter value can be edited

_Type	
Type	Counter
Detail	
Coefficient	10
Counter Max	999999
Default Value	0
Init Value	1
Mandatory	0
OverWrite	0
Phys. Units	kWh
Tag Name	P14-P141-B118592.CV
Validated	0
Write Value	0
General	
Comment	Counter example
Name	P14-P141-B118592
Location	
Column	1
Line	2

FIGURE 24 - COUNTER FIELD PROPERTIES

Formula

A Formula field allows to calculate a resulting value from operations on other input data.

Avg of 50FI0070 and 50FI0072 (measured)		t/h

FIGURE 25 - FORMULA FIELD

A Formula field has the following properties:

- Formula: list of operations that will be defined. To open the formula editor, click on the  button that appears when clicking on (Collection), or right click on the control and select properties
- Phys. Units: physical units that will be associated with the result of the formula

Type	
Type	Formula
Detail	
Formula	(Collection)
Phys. Units	t/h
General	
Comment	Avg of 50FI0070 and 50FI0072 (measured)
Name	AVG_test
Location	
Column	1
Line	24

FIGURE 26 - FIELD PROPERTIES

The formula is calculated through a succession of elementary operation (+,-,*,/) applied to two variables, which can refer to any field value, including other formula, in any section. It can also refer directly to a numeric value or to the result of an elementary operation of the formula itself.

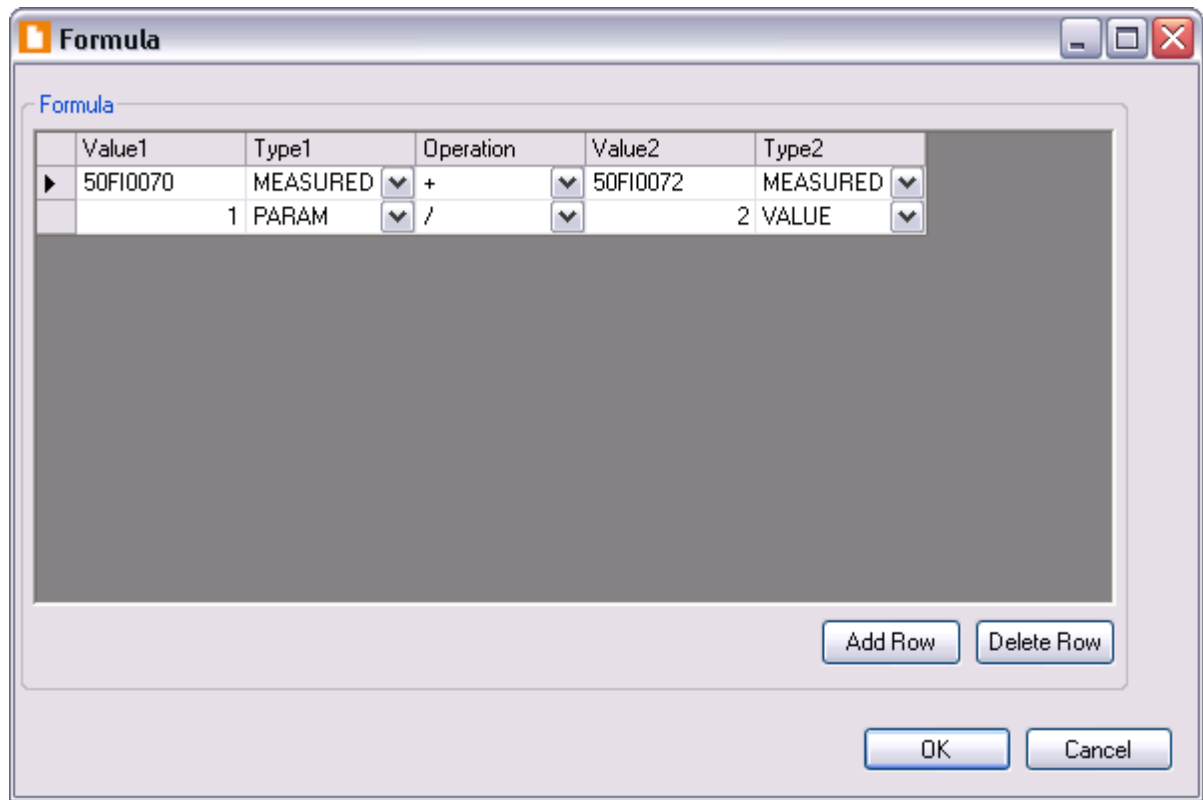


FIGURE 27 - DEFINING A FORMULA (AVERAGE VALUE OF TWO FLOW RATES)

Each variable is defined by two fields: Type and Value.

TYPE	VALUE	DESCRIPT ION
MEASURED	Tag name	Measured value of a tag
VALIDATED	Tag name	Validated value of a tag
USED	Tag name	Used value (after overwritin g) of a tag
TANK_MEASURED	Tank tag name	Measured final inventory

TYPE	VALUE	DESCRIPT ION
TANK_USED	Tank tag name	Used value (after overwritin g) of a tank final inventory
TANK_DELTA	Tank tag name	Delta inventory of a tank
TANK_VALIDATED	Tank tag name	Validated value of a tank final inventory
COUNTER_CONVERTED	Counter name	Converted value of a counter
COUNTER_USED	Counter name	Used value (after overwritin g) of a counter
COUNTER_VALIDATED	Counter name	Validated value of a counter
TABLE	Tag name	Value from a TableInput field
FORMULA	Formula name	Result of an other formula
VALUE	Numeric value	A numeric value
PARAM	Row number	Result of

TYPE	VALUE	DESCRIPT ION
		another line of the formula's list

TABLE 1 - VARIABLES AVAILABLE IN A FORMULA

If a tag value is used, it is mandatory that a Tag Input field is created on this tag.

Free Input

A Free Input field enables to enter a freely formatted text into the measurement file. The only constraint is of course that this text complies with the various syntaxes available in measurement files.



FIGURE 28 - FREE INPUT FIELD

A Free Input field does not have particular properties.

The name of the field may not include any space.

Product

A Product field enables to allocate a product to a Vali unit (tank) or stream.

Operation Mode	011-CDU_STATU	IHDSDIE	▼

FIGURE 29 - PRODUCT FIELD PROPERTIES

A Product field has the following properties:

- Default Value: default value for the product
- DisplayName: product list name will be displayed
- Mandatory: a valid value must be given to the switch to approve the data
- Product: specifies the product list from which the product must be selected. See [Products lists](#) section to define products list. The list of available products depends on the list that is chosen
- Product Type: type: U for Unit , S for Stream, T for Tag
- Unit/Stream/Tag: Unit, Stream or Tag name

Type	
Type	Product
Detail	
Default Value	IHDSDIE
Display Name	1
Mandatory	0
Product	BLD-DSL
Product type	T
Unit/Stream/Tag	661-T-1004-IN_F
General	
Comment	Closing Product
Name	T_661-T-1004-IN
Location	
Column	1
Row	3

FIGURE 30 - FIELD PROPERTIES

Status

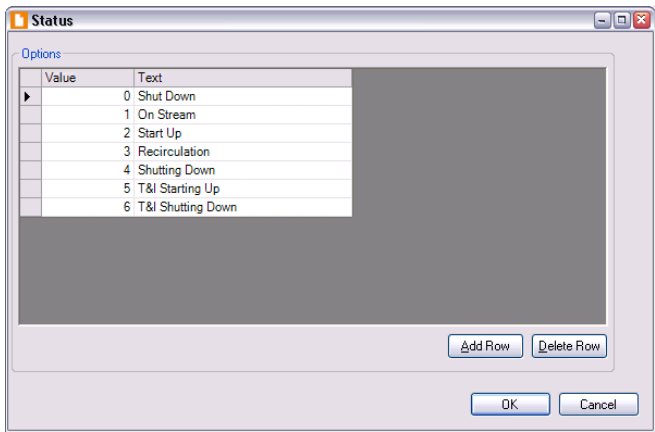
A Status field enables to assign different values to a tag according to a pre-defined status.



FIGURE 31 - STATUS FIELD

A switch field has the following properties:

- Default Value: default value for the switch, to be selected from the list of values. The default value will be used if not value is retrieved from the mea file associated with the section
- Mandatory: a valid value must be given to the switch to approve the data
- Options: possible values for the status. To edit the list of options, click on .



Type	
Type	Status
Detail	
Default Value	1
Mandatory	0
Options	(Collection)
General	
Comment	Operation Mode
Name	011-CDU_STATU
Location	
Column	1
Row	1

FIGURE 32 - FIELD PROPERTIES

Switch

A Switch field enables to enter pre-configured mea file commands into the mea file according to a user selection between a number of values for the switch.

Switch example	SWT-U31	Option 1	▼

FIGURE 33 - SWITCH FIELD

A switch field has the following properties:

- **Default Value:** default value for the switch, to be selected from the list of values. The default value will be used if not value is retrieved from the mea file associated with the section
- **Mandatory:** a valid value must be given to the switch to approve the data
- **Options:** possible values for the switch. For each of these values, a number of lines can be defined in the table on the right of the list.

_Type	
Type	Switch
Detail	
Default Value	Option 1
Mandatory	0
Options	(Collection)
General	
Comment	Switch example
Name	SWT-U31
Location	
Column	1
Line	11

FIGURE 34 - FIELD PROPERTIES

Table

A Table field enables to enter input data in a table form, like for example the amount of materials used in various operation units.

	Column1	Column2	Total
► 50FI0070			
50FI0072			
Total			

FIGURE 35 - TABLE FIELD

The input data will be associated to tags that will be defined manually by entering a tag name in a cell or automatically using an auto-naming strategy. In the first case, only cells associated with a tag will be editable in ValiForms, and the measured value of the tag will be retrieved. In the second, all cells are editable.

A table field has the following properties:

- Auto Name: defines how to automatically name the tags for each cell (%1 and %2 stand for the row and column names)
- Auto Size: auto-size columns when required, i.e. if the number of columns can not fit in the window width. If set to false, a scrollbar appears at the bottom of the table
- Decimals: maximum number of decimals
- Decimals in Table: numbers must be truncated to the number of decimals defined in Decimals
- Default Value: default value for the cells
- Phys. Units: physical units
- Total Row: an additional column is created before or after the table with the sum of the rows
- Total Column: an additional row is created before or after the table with the sum of the columns

_Type	
Type	TableInput
Detail	
Auto name	
Auto Size	0
Decimals	2
Decimals in Table	True
Default Value	0
Phys. Units	
Total Column	1
Total Row	1
General	
Comment	
Name	
Location	
Column	1
Line	12

FIGURE 36 - TABLE FIELD PROPERTIES

TagCalc

A TagCalc field allows to recompute a value of a tag based in another tag, using a multiplying coefficient

TagCalc example	54FI1012				t/h

FIGURE 37 - TAGCALC FIELD PROPERTIES

A TagCalc field has the following properties:

- Coefficient: multiplier for the calculation
- Default Value: default value for the input data
- Mandatory: a valid value must be provided to approve the data
- OverWrite: the measured value can be overwritten by the user
- Phys. Units: physical units (use =BLS to select the units form the model)
- TagName: Name of the tag to be calculated
- WriteValue: the measured tag value can be edited

[-] _Type	
Type	TagCalc
[-] Detail	
Coefficient	3
Default Value	3.795
Mandatory	0
OverWrite	1
Phys. Units	t/h
Tag Name	54FI1012
Write Value	0
[-] General	
Comment	TagCalc example
Name	54FI1012
[-] Location	
Column	1
Line	17

FIGURE 38 - FIELD PROPERTIES

TagFormula

A TagFormula field allows to calculate three consecutive resulting values from operations on other input data.

Total Inlet				t/h

FIGURE 39 - TAGFORMULA FIELD

A TagFormula field has the following properties:

- Formula1: list of operations that will be defined in the first cell. To open the formula editor, click on the  button that appears when clicking on (Collection), or right click on the control and select properties
- Formula2: list of operations that will be defined in the second cell. To open the formula editor, click on the  button that appears when clicking on (Collection), or right click on the control and select properties
- Formula3: list of operations that will be defined in the second cell. To open the formula editor, click on the  button that appears when clicking on (Collection), or right click on the control and select properties
- Phys. Units: physical units that will be associated with the result of the formula

Type	
Type	TagFormula
Detail	
Formula1	(Collection)
Formula2	(Collection)
Formula3	(Collection)
Phys. Units	t/h
General	
Comment	Total Inlet
Name	011-CDU Inlet
Location	
Column	1
Row	34

FIGURE 40 - FIELD PROPERTIES

The formula is calculated through a succession of elementary operation (+,-,*,/) applied to two variables, which can refer to any field value, including other formula, in any section. It can also refer directly to a numeric value or to the result of an elementary operation of the formula itself.

Value1	Type1	Operation	Value2	Type2
50FI0070	MEASURED	+	50FI0072	MEASURED
1	PARAM	/	2	VALUE

Add Row Delete Row

OK Cancel

FIGURE 41 - DEFINING A FORMULA (AVERAGE VALUE OF TWO FLOW RATES)

Each variable is defined by two fields: Type and Value.

TYPE	VALUE	DESCRIPTION
MEASURED	Tag name	Measured value of a tag
VALIDATED	Tag name	Validated value of a tag
USED	Tag name	Used value (after overwriting) of a tag
TANK_MEASURED	Tank tag name	Measured final inventory

TYPE	VALUE	DESCRIPT ION
TANK_USED	Tank tag name	Used value (after overwritin g) of a tank final inventory
TANK_DELTA	Tank tag name	Delta inventory of a tank
TANK_VALIDATED	Tank tag name	Validated value of a tank final inventory
COUNTER_CONVERTED	Counter name	Converted value of a counter
COUNTER_USED	Counter name	Used value (after overwritin g) of a counter
COUNTER_VALIDATED	Counter name	Validated value of a counter
TABLE	Tag name	Value from a TableInput field
FORMULA	Formula name	Result of an other formula
VALUE	Numeric value	A numeric value
PARAM	Row number	Result of

TYPE	VALUE	DESCRIPT ION
		another line of the formula's list

TABLE 2 - VARIABLES AVAILABLE IN A FORMULA

If a tag value is used, it is mandatory that a Tag Input field is created on this tag.

TagInput

A Tag Input field allows to change measured value of a tag.

Tag input example	54FI1012.TI				

FIGURE 42 - TAGINPUT FIELD PROPERTIES

A TagInput field has the following properties:

- Calculated Value: the tag value will be calculated according to a formula
- Default Value: default value for the tag
- Formula: define a formula to calculate a tag value. Use of formula is described in section [Formula](#)
- Mandatory: a valid value must be provided to approve the data
- Max: maximum value for the measured and overwrite values (selecting =H,=HH or =SH will select the corresponding bound from the bls model file)
- Min: minimum value for the measured and overwrite values (selecting =L,=LL or =SL will select the corresponding bound from the bls model file)
- OverWrite: the measured value can be overwritten by the user
- Phys. Units: physical units (use =BLS to select the units form the model)
- Validated: the validated value will be shown
- WriteValue: the measured tag value can be edited

_Type	
Type	TagInput
Detail	
Calculated Value	False
Default Value	-9999
Formula	(Collection)
Mandatory	0
Max	
Min	
OverWrite	1
Phys. Units	
Validated	1
Write Value	0
General	
Comment	Tag input exampl
Name	54FI1012.TI
Location	
Column	1
Line	19

FIGURE 43 - FIELD PROPERTIES

Tank



FIGURE 44 - TANK FIELD PROPERTIES

A Tank field has the following properties:

- Default Value: default value for the tag
- Imb Mode: a check box will enable the user to set the tank in "imbalance mode"
- Init Value: an initial value will be read from the aux. mea file or from ValiDatabase. Otherwise, the initial value is set to 0
- Mandatory: a valid value must be provided to approve the data
- OverWrite: the measured value can be overwritten by the user
- Phys. Units: physical units (use =BLS to select the units form the model)
- Tag Name: not used
- Validated: the validated value will be shown
- WriteValue: the measured tag value can be edited
- Tank Name: name of the Tank

Type	
Type	Tank
Detail	
Default Value	-9999
Imb Mode	False
Init Value	0
Mandatory	0
OverWrite	0
Phys. Units	
Tag Name	
Validated	0
Write Value	0
General	
Comment	Tank example
Name	
Tank Name	
Location	
Column	1
Row	21

FIGURE 45 - TANK FIELD PROPERTIES

2.1.3 CONFIGURING AN OUTPUT SECTION

A field can be placed on any free space of the grid. The grid can be showed/hidden using keyboard shortcut ctrl+g or by selecting **View | Grid**.

The different fields are:

- Calculated: value calculated from other input or output data
- PhysUnit: gets the physical units of a tag
- TagBlock: block of tag values following the same display format
- TagText: text associated to a tag
- TagValue: individual tag value
- TankBlock: block of tank values following the same display format
- Text Field: text to be displayed in the Form
- Title: title text (underlined) to be displayed in the Form
- Trend: historical trend of tag value

PROPERTIES OF AN OUTPUT SECTION

An output section allows to define output sub-sections.

An output section has the following properties:

- Aggreg.mode: default aggregation mode of the section, used as default when creating new sub-sections
- Approval: specifies that the data must be approved
- Name: section name displayed in the main tabs of the Form
- Overwrite With Historian: measurements as available in the main Mea file is always overwritten by the data from the historian file

General	
Aggreg. Mode	VALUE_BEGINDA
Approval	False
Name	Output full
Overwrite With	False
Style	Output

FIGURE 46 - OUTPUT SECTION PROPERTIES

PROPERTIES OF AN OUTPUT SUB-SECTION

An output sub-section allows to define output fields.

An output sub-section has the following properties:

- Columns: the maximum number of columns used in the report
- Aggreg.Mode: default aggregation mode of the sub-section
- Initial Value: specifies whether the initial counter and tank inventory data must be taken on the preceding DAILY or MONTHLY period
- Name: sub-section name displayed in the secondary tabs of the section
- Show Date: specifies whether the run date must be shown at the top of the sub-section. When aggregating data, the begin and end dates of the aggregation period will be displayed
- Show Quality: specifies whether a run quality indicator must be displayed at the top of the sub-section. (Red: quality > 2.5, Green: quality < 1.5, or orange otherwise). When aggregating data, the quality indicator is not shown

Detail	
Columns	14
General	
Aggreg. Mode	
Initial Value	DAILY
Name	SubSection
Show Date	False
Show Quality	False

FIGURE 47 - OUTPUT SUB-SECTION PROPERTIES

COMMON PROPERTIES

Following properties are common to all fields:

- Comment: some text that is displayed in front of the control (not used for title fields)
- Name: name of the control, also used as name of the tag associated with the control (tag for which value will be retrieved, or tag for which output will be written). This property is mandatory for Free Input field
- Location: the location of the control on the grid, indicated by the row and column number

All fields can be duplicated. To duplicate a field, right click it and select Duplicate.

THE DIFFERENT FIELDS

Calculated value

A Calculated Value field calculates a value from a formula, using values from other cells in the ValiForms sheet.



FIGURE 48 - CALCULATED VALUE FIELD

A Calculated Value field has the following properties:

- Mode: operation: SUM, DIF, DIV and MULT
- Range: range of values on which operation is applied. An element is defined by its location "row.column". The separator to use is a dot(.)

Example: Range 4.2 represents the value located in row 4 and column 2

To define a range of values, the first and the last location are used. The separator to use is a dash (-).

Example: Range 4.2-6.2 represent these values: 4.2, 5.2 and 6.2

To define more than one range of values, one of the separators comma(,) or semicolon(;) is to be used.

Example: Range 4.2-6.2;4.3-5.3;4.4 represent these values: 4.2, 5.2, 6.2, 4.3, 5.3 and 4.4

Type	
Type	CalculatedValue
Detail	
Mode	SUM
Range	4.2;6.1
General	
Comment	
Name	
Location	
Column	5
Row	1
Size	
Height	1
Width	1

FIGURE 49 - CALCULATED VALUE FIELD PROPERTIES

Phys Unit

A Phys Unit field displays the physical units of a tag.

PhysUnit

FIGURE 50 - PHYS UNIT FIELD

A Phys Unit field has the following properties:

- Width: width of the field

▣	Type	
Type	PhysUnit	
▣	General	
Comment	PhysUnit	
Name	50FI0070	
▣	Location	
Column	3	
Line	1	
▣	Size	
Width	1	

FIGURE 51 - PHYS UNIT FIELD PROPERTIES

TagBlock

A TagBlock field lists various properties of several tags in a single block.

24FI0004	0	0	
00JI1000	0	0	
20FI0006	0	0	
24FI0001	0	0	
24FI0012	0	0	

FIGURE 52 - TAGBLOCK FIELD

The first 4 columns define various settings of a selected tag. You can add columns to display various values of these tags (none, comment, measured, used or validated value or accuracy, penalty)

A TagBlock field has the following properties:

- Tag Block: list of tags and properties to display

▣	_Type	
Type	TagBlock	
▣	Detail	
Tag Block	(Collection)	
▣	General	
Comment		
Name	TagBlock	
▣	Location	
Column	1	
Line	3	
▣	Size	
Width	5	

FIGURE 53 - TAGBLOCK FIELD PROPERTIES

Click on the  button that appears when clicking on (Collection) to select tags and properties to display:

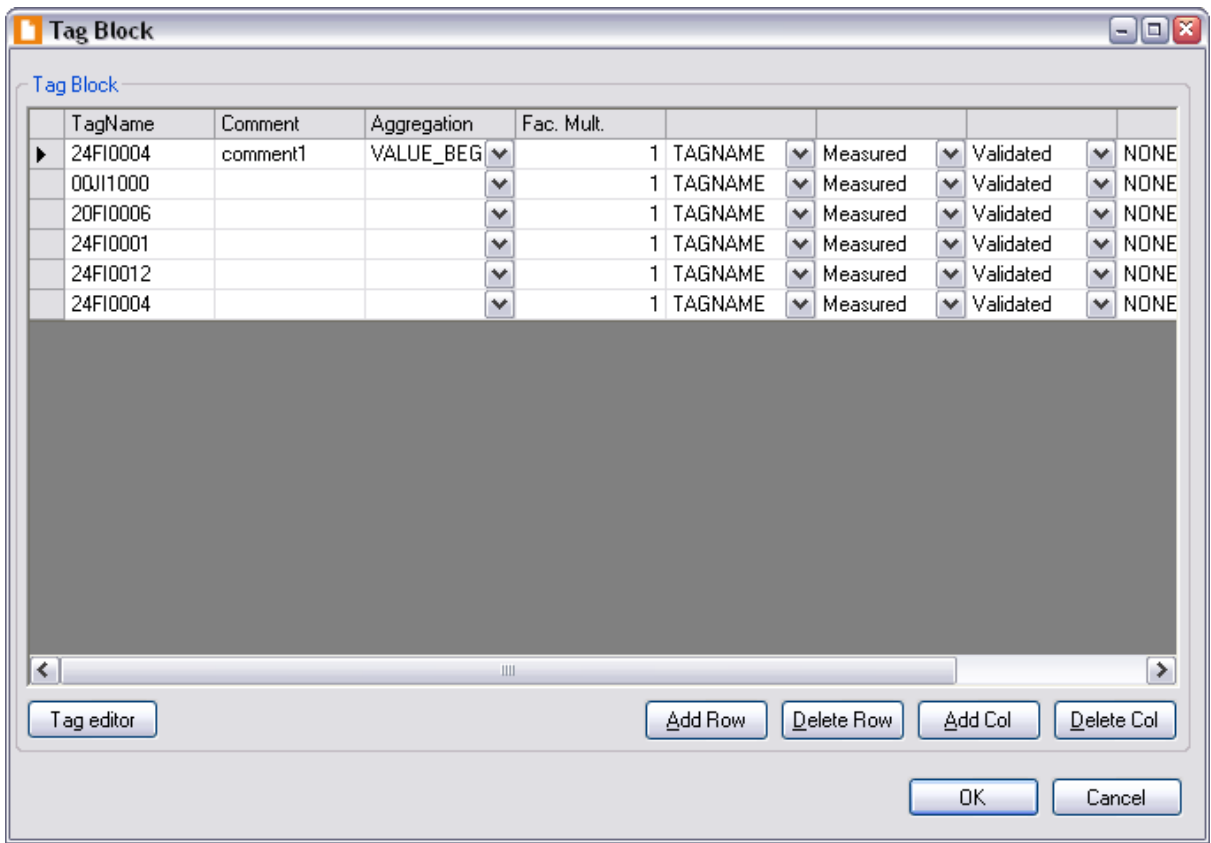


FIGURE 54 - TAGBLOCK EDITOR

The first four columns define general properties:

- TagName: name of the tag
- Comment: comment associated to the tag
- Aggregation: aggregation mode for the tag
- Fac. Mult: multiplier for the tag values

Rows can be added and deleted using buttons located at the bottom right of the table.

To add properties, select Add Col, and then select the property to display in the drop-down menu (TAGNAME, COMMENT, NONE, Measured, Validated, Penalty, Mea.Acc., Mea.Fil., Val.Acc., Acc.Fil., PhysUnit).

Tags for which information will be displayed can be chosen using the tag editor button:

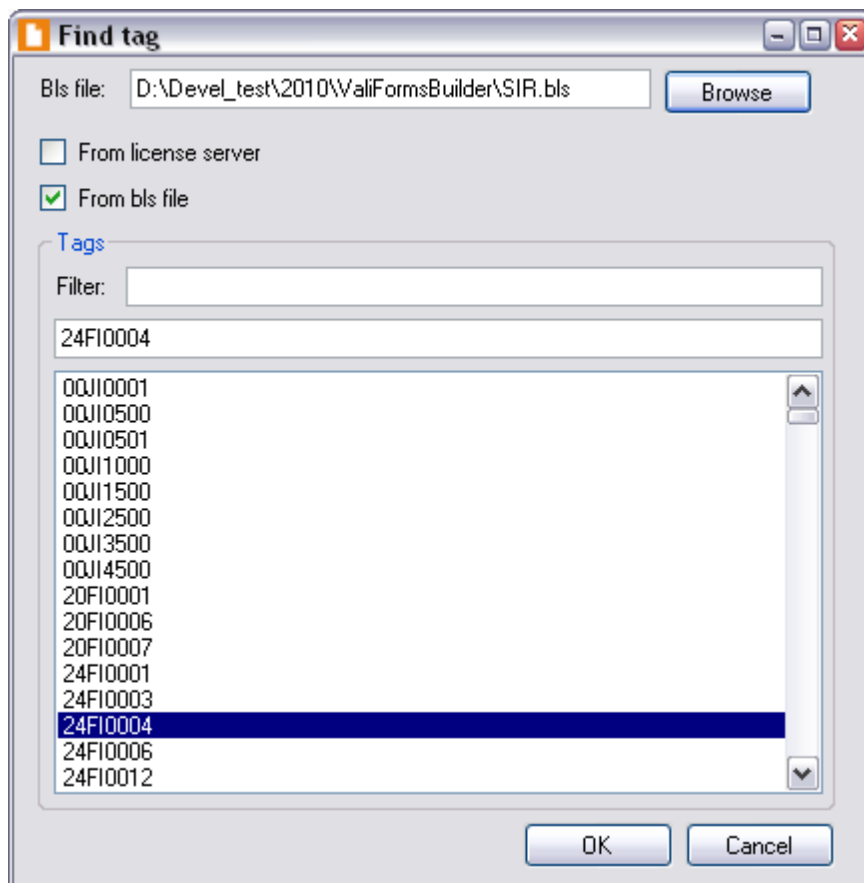


FIGURE 55 - TAG EDITOR

This window allows you to select the tag name from a bls file or from your LicenseServer.

TagText

A TagText Field displays general information associated to a run: file name, model or main PFD name, model or VALI engine version, any INFOi field.

TagValue

A TagValue Field displays a single value from a tag.

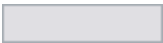


FIGURE 56 - TAGVALUE FIELD

A TagValue field has the following properties:

- Fac. Mult: multiplier for the tag values
- Mode: aggregation mode
- Value: value to be displayed: Acc.Fil., Compensated, Eliminated, Filtered, Gain, In Use, Measured, Mea.Acc., Mea.Fil, Penalty, Status, Used Penalty, Validated, Val.Acc., VDI Penalty
- Height: height of the field (in number of rows)
- Width: width of the field (in number of columns)

_Type	
Type	TagValue
Detail	
Fac. Mult.	1
Mode	VALUE_BEGINDA
Value	MEASUREMENT
General	
Comment	Tag Value
Name	24FI0004
Location	
Column	1
Line	10
Size	
Height	1
Width	1

FIGURE 57 - TAGVALUE FIELD PROPERTIES

TankBlock

A TankBlock field lists various properties of several storage tanks in a single block.

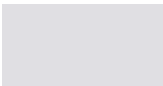


FIGURE 58 - TANK BLOCK FIELD PROPERTIES

A TankBlock field has the following properties:

- Show Comment: display the comment value
- Show Delta Valid. Value: display the delta validated value
- Show Delta Value: display the delta value
- Show Final Value: display the final value
- Show Initial Value: display the initial value
- Show Overwrite Value: display the overwrite value
- Show Phys. Unit: display the physical unit value
- Show Tag Name: display the tag name value
- Show Validated Value: display the validated value
- Tank Block: allows to define list of tanks

Type	
Type	TankBlock
Detail	
Show Comment	False
Show Delta Valid. Value	False
Show Delta Value	False
Show Final Value	False
Show Initial Value	False
Show Overwrite Value	False
Show Phys. Unit	False
Show Tag Name	False
Show Validated Value	False
Tank Block	(Collection)
General	
Comment	Tank Block
Name	TankBlock
Location	
Column	1
Row	11

FIGURE 59 - TANK BLOCK FIELD PROPERTIES

Click on the  button that appears when clicking on (Collection) to select tags to display:

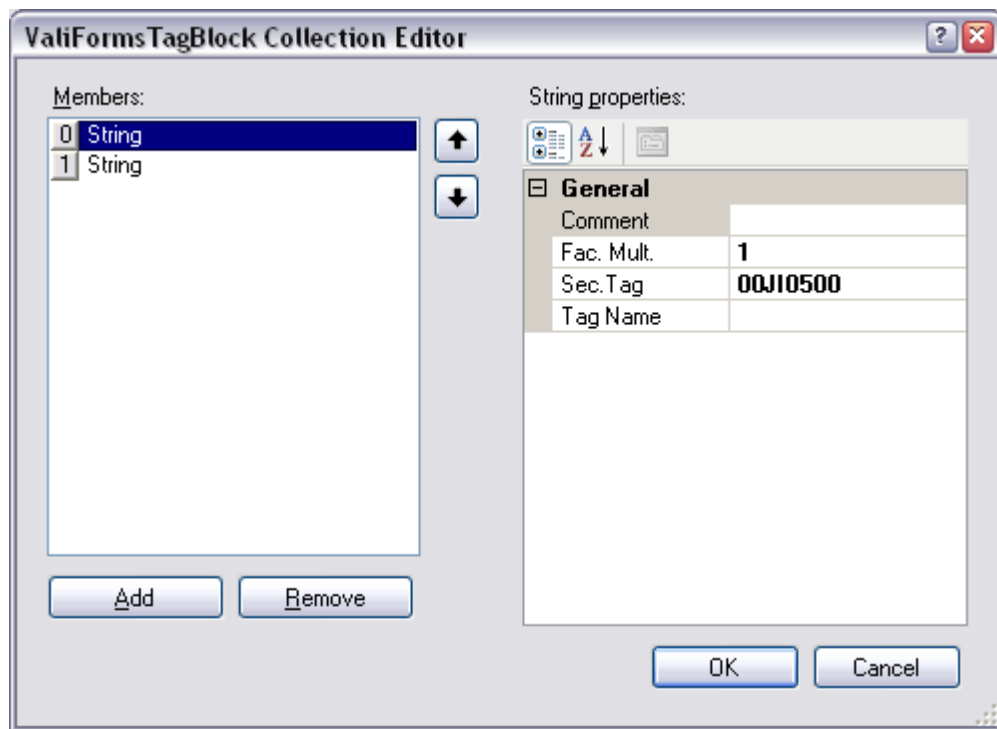


FIGURE 60 - TANK BLOCK EDITOR

This window allows you to configure tags for tank monitoring by setting:

- Comment: comment associated to the tag
- Fac. Mult: multiplying factor for the tank values
- Sec.Tag: name of the tag giving the initial inventory
- TagName: name of the tag giving the final inventory

Text Field

Text field is used to enter free text on the form.

Trend

A Trend field displays historical trends of selected tag values.

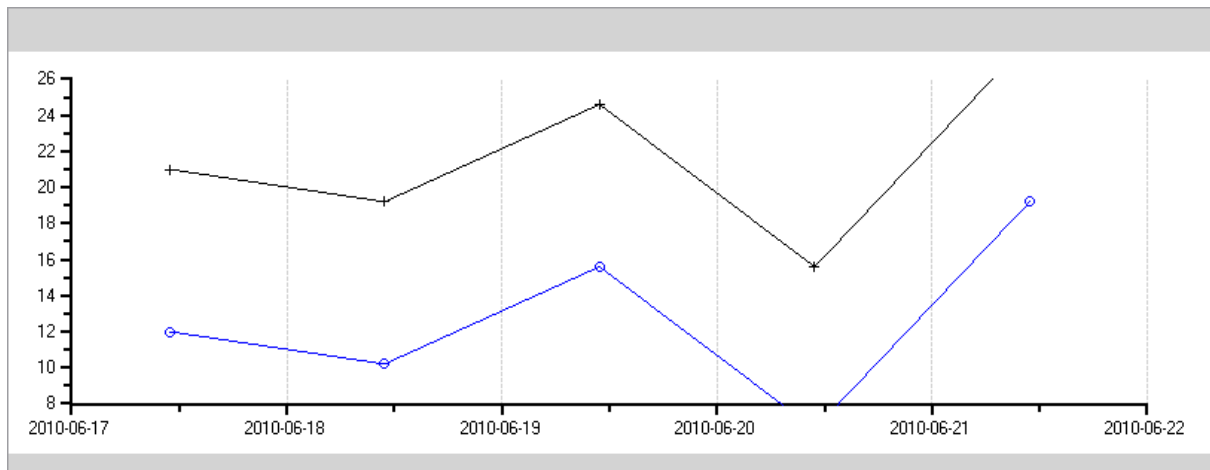


FIGURE 61 - TREND FIELD

A Trend field has the following properties:

- Axis:
 - Grid X/Y: shows/hides grid on horizontal/vertical axis
 - Label X/Y: label of the axis
 - Min/Max Y: minimum/maximum value of the vertical axis
 - Min/Max Y Enabled: force use of minimum/maximum values entered. if set to false, minimum/maximum value will be adjusted automatically
 - Visible X/Y: shows/hides horizontal/vertical axis
- Chart area
 - Color: background color of the trend area
 - Maximized: maximized or not
- General
 - **Comment**
 - **Mode: aggregation mode**
 - **Name**
- Plot:
 - Background: background color of the plot area
 -
- Tags:
 - Data: the collection of tags that will be displayed. Click on the  button that appears when clicking on (Collection), or right click on the trend and select properties to select data to display.
- Trend:
 - Background: background color of the field
 - Layout: horizontal or vertical
 - Show legend: shows/hides the legend of the trend
 - Time Range: DAY, WEEK, MONTH, YEAR
 - Time Unit: Number of periods in the trend

Type	
Type	Trend
Axis	
Grid X	True
Grid Y	False
Label X	
Label Y	
Max Y	10
Max Y Enabled	False
Min Y	0
Min Y Enabled	False
Visible X	True
Visible Y	True
Chart area	
Color	Aquamarine
Maximized	True
General	
Comment	some comment
Mode	VALUE_BEGIN
Name	Trend
Location	
Column	1
Row	15
Plot	
Background	AntiqueWhite
Size	
Height	15
Width	10
Tags	
Data	{Collection}
Trend	
Background	LightGray
Layout	Vertical
Show legend	False
Time Range	DAY
Time Unit	-7
Title	
Type	XYPlot

FIGURE 62 - TREND FIELD PROPERTIES

When clicking on the  button that appears when clicking on (Collection), or right clicking on the trend and select properties, data to display can be selected:

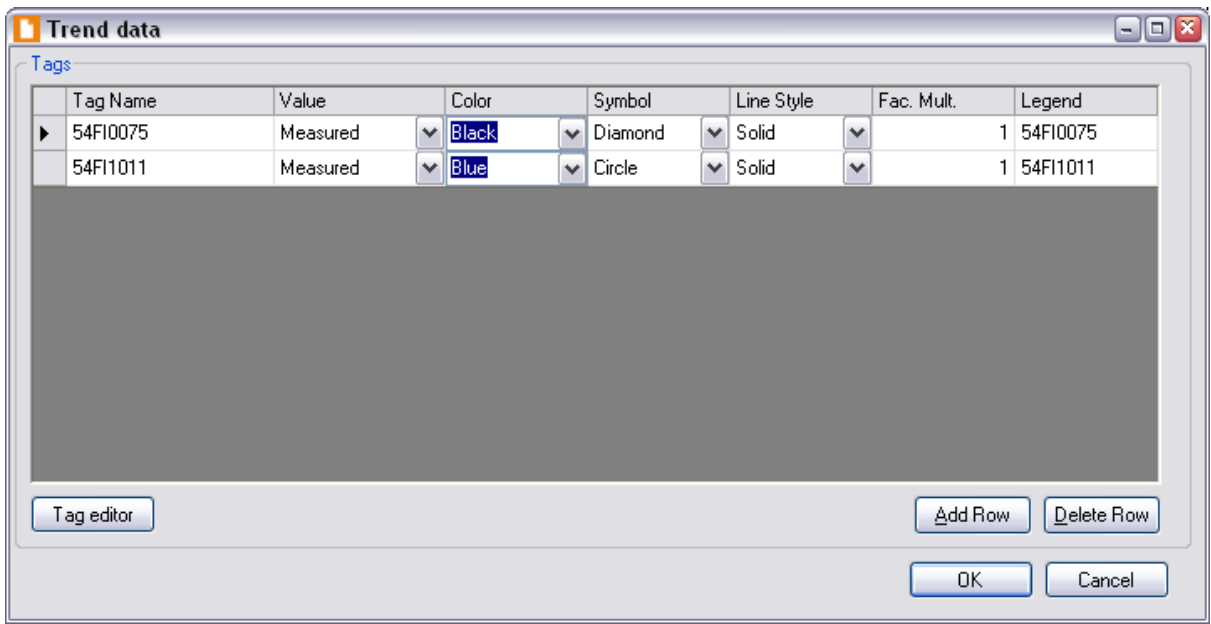


FIGURE 63 - TREND DATA

This window allows to choose the tag to display, which value to trend (Acc.Fil., Compensated, Eliminated, Filtered, Gain, In Use, Measured, Mea.Acc., Mea.Fil, Penalty, Status, Used Penalty, Validated, Val.Acc., VDI Penalty), the color and the style of the lines and symbols, a multiplier for the value and a legend.

SECTION 3: USING VALIFORMS

3.1 STARTING VALIFORMS

To start ValiForms in interactive mode:

- double click the application ValiForms.exe in the bin folder of VALI installation
- select the configuration xml file
- the ValiForms window is then displayed

To start ValiForms in background mode, start ValiForms in <installdir>\bin folder (by default C:\Program Files\Belsim\bin or C:\Belsim\bin) with the following mandatory arguments:

- the keyword -BG
- the run date (in format yyyyMMdd_HH:mm:ss, optionally surrounded by double quotes)
- the duration value and its physical units (**Hours|H|Min|D|Day**)
- the name of the configuration xml file

Optionally you can use the following arguments:

- the keyword -A followed by an aggregation period:
 - DD: Day to Date
 - WD: Week to Date
 - MD: Month to Date
 - YD: Year to Date
 - LD: Last Day
 - LW: Last Week
 - LM: Last Month
 - LY: Last Year
- the keyword -I (or -O) to print only the input (or output) section. By default, only the output section is printed

Examples:

```
C:\Program Files\Belsim\bin\ValiForms.exe -BG 20110314_000000 1 H "D:\Tests\ValiForms\test.xml"
C:\Program Files\Belsim\bin\ValiForms.exe -BG 20110314_000000 1 H "D:\Tests\ValiForms\test.xml" -I
C:\Program Files\Belsim\bin\ValiForms.exe -BG "20110314_000000" 1 H "D:\Tests\ValiForms\test.xml" -A LW
```

Attention:

1. You can also start ValiForms using a short-cut pointing to the application.
2. You can provide the name of the xml file as argument to the application.

3. In a ValiProduction application, ValiForms can be called directly from ValiManager, opening the section corresponding to active run and step.
4. In a ValiPerformance application, ValiForms can be called directly from ValiService/ ValiScheduler to generate (and possibly e-mail) reports. In such a case, you only need to specify the xml file name and path and optionally use the options -I, -O and -A.

3.2 USING VALIFORMS IN INTERACTIVE MODE

ValiForms - ValiProduction

Area 1 Area 2 Area 3 Area 4 Area 5 Options

Unit 1 Raw Materials Unit 2/3

Date:

Stockages

Description	Tag Name	Initial	Final	Corrected	Validated	Delta	Delta Val	Eng Units	Imbalance
Storage T1	1LMB4271_BI.TJ	0						TON	☐
Storage T2	1LMB4352_BI.TJ	0						TON	☐
Storage T3	1LMB4702_BI.TJ	0						TON	
Storage T4	7LMB3011_BI.TJ	0						TON	☐
Storage T5	9LMB465_BI.TJ_	0						TON	☐
Storage T6	9LMB466_BI.TJ_	0						TON	

Production

Description	Tag Name	Measured	Corrected	Validated	Eng Units
Third party	9FMB8011_BI.TJ	-9999			TON/D
Product 1 97% -> T4	1FMB4721.TJ_1	-9999			M3/D
Product 1 95% -> T5	7FMB3031.TJ_1	-9999			M3/D

Consumption

Description	Tag Name	Measured	Corrected	Validated	Eng Units
Product 1 97% -> Unit 2	1FMB1211.TJ_1	-9999			TON/D
Product 1 95% -> Unit 2	1FMB1213.TJ_1	-9999			TON/D
Product 1 97% -> Unit 3	1FMB2211.TJ_1	-9999			TON/D
Product 1 95% -> Unit 3	1FMB2213.TJ_1	-9999			TON/D
Product 1 -> Unit 2/3	SWT-NAM	Product 1 -> Unit 2			
	1FMB1215.TJ_1	-9999			TON/D
Loading 93%	NAM-C93N_MAS	0			TON/D
Condensat	1FMB4603.TJ_1	-9999			TON/D

Dilution Water

Description	Tag Name	Measured	Corrected	Validated	Eng Units
Dilution 97 -> 95%	1FMB4722.TJ_1	-9999			M3/D
Dilution 95 -> 93%	7FMB3111.TJ_1	-9999			M3/D

☐ Aggregation

Ready

FIGURE 64 - A TYPICAL VALIFORMS INPUT FORM

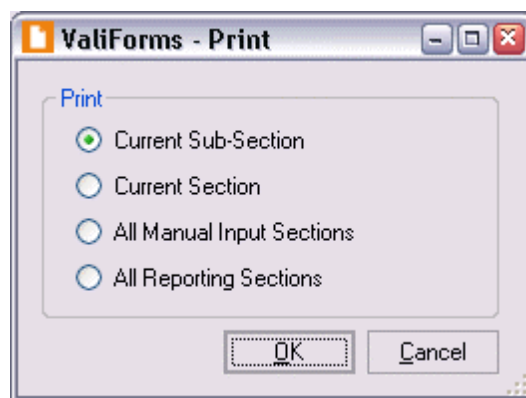
Depending on the configuration of the xml file, some of the options described below may be deactivated. For example, the aggregation and date browsing option will only be operational when in connection with a ValiDatabase or with a ValiProduction application.

The user can only enter data in the blank fields. All other fields are either static data, read-only data, calculated or validated values.

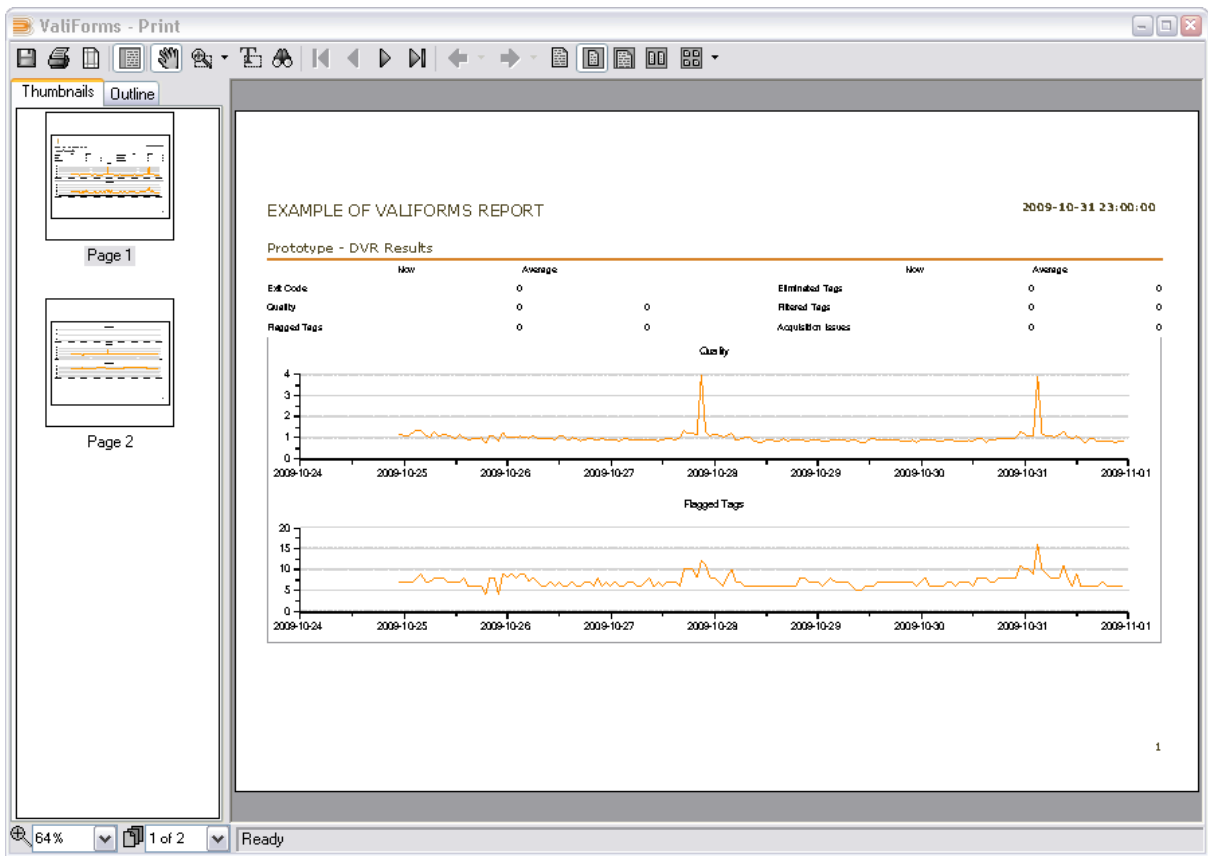
Following buttons are available:

- Aggregation: aggregate data from consecutive runs between two selected dates
- Print: prints the Form contents (see more information below)
- Save: saves the input data
- Save and Approve: saves the input data and sets the step as being completed (available only when ValiForms is run from a ValiProduction application)
- About: display Belsim coordinates
- Help: activate the Help file
- Close: closes the application. Take care that input data are not saved when closing the application

The Print command activates first a selection screen to specify what needs to be printed out.



A preview of the printed report is then displayed.



The menu bar gives access to the following options:

-  : save the report as a pdf, xls, emf or rtf file
-  : print the report
-  : change the page set-up
-  : toggle the navigation bar
-  : activate the hand tool, to move the report in the preview window
-  : activate the zoom tool, to zoom in or out the preview window
-  : select text in the report
-  : search text in the report
-  : go back to the first page
-  : go to previous page
-  : go to next page
-  : go to last page
-  : go to preceding view
-  : go to next view
-  : preview in actual size
-  : preview a full page
-  : preview with actual page width
-  : preview two pages
-  : preview a selected number of pages



belsim

CHAPTER 18

THERMO

SECTION 1: INTRODUCTION

Thermo is a tool dedicated to thermodynamic calculations, such as:

- thermodynamic properties calculations (enthalpy, entropy, etc.)
- thermodynamic and state diagrams
- phase equilibrium at constant pressure or temperature
- thermodynamic equilibrium using the White method

Thermo tool offers various benefits as it allows to:

- initialize the starting point of a reconciliation problem
- analyze the evolution of the properties of a given stream
- determine the dew/bubble point as well as the vapor fraction of a mixture

Thermo handles calculations for both **pure components** and **mixtures**. The components can either be defined by the USER or loaded from the BELSIM database which contains more than 830 compounds (including acetylene, acids, alcohols, aromatics, olefins etc.). Pseudo-compounds are also available for petroleum fractions, fuel-oil and coal modeling.

The Thermodynamic methods are

- ideal gas/liquid
- Peng-Robinson
- Soave
- Lee-Kesler-Plöcker
- LASSC, Redlich-Kwong
- Curl-Pitzer
- Virial equation of state
- NBS, VDI and IAPWS-IF97 models for steam/water
- NRTL
- UNIQUAC
- UNIFAC
- Chao-Seader
- Grayson-Streed
- Braun K10

Please refer to Section 3 of the ValiModeller help file for more information on the thermodynamic

methods.

The necessary input for Thermo is a BELSIM model file <model.bls>. Please refer to Section 4 of the ValiModeller help file on how to build model files. The model file defines:

- the composition of the streams
- the stream thermod, i.e. the group of thermodynamic methods that are defined on the stream

Thermodynamic calculations are performed within the library Thermo4.dll which is the core engine of Thermo while Thermo.exe is used for the graphical user interface.

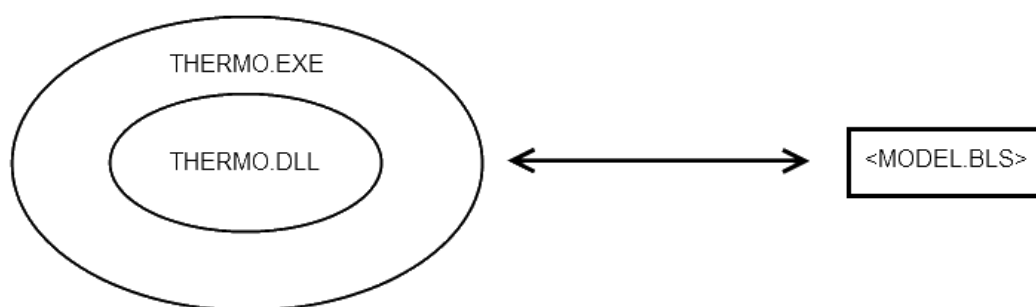


FIGURE 1 - THERMO ARCHITECTURE

SECTION 2: USING THERMO

2.1 STARTING THERMO

To start Thermo go to the bin directory of the VALI installation folder and double-click on "Thermo.exe".

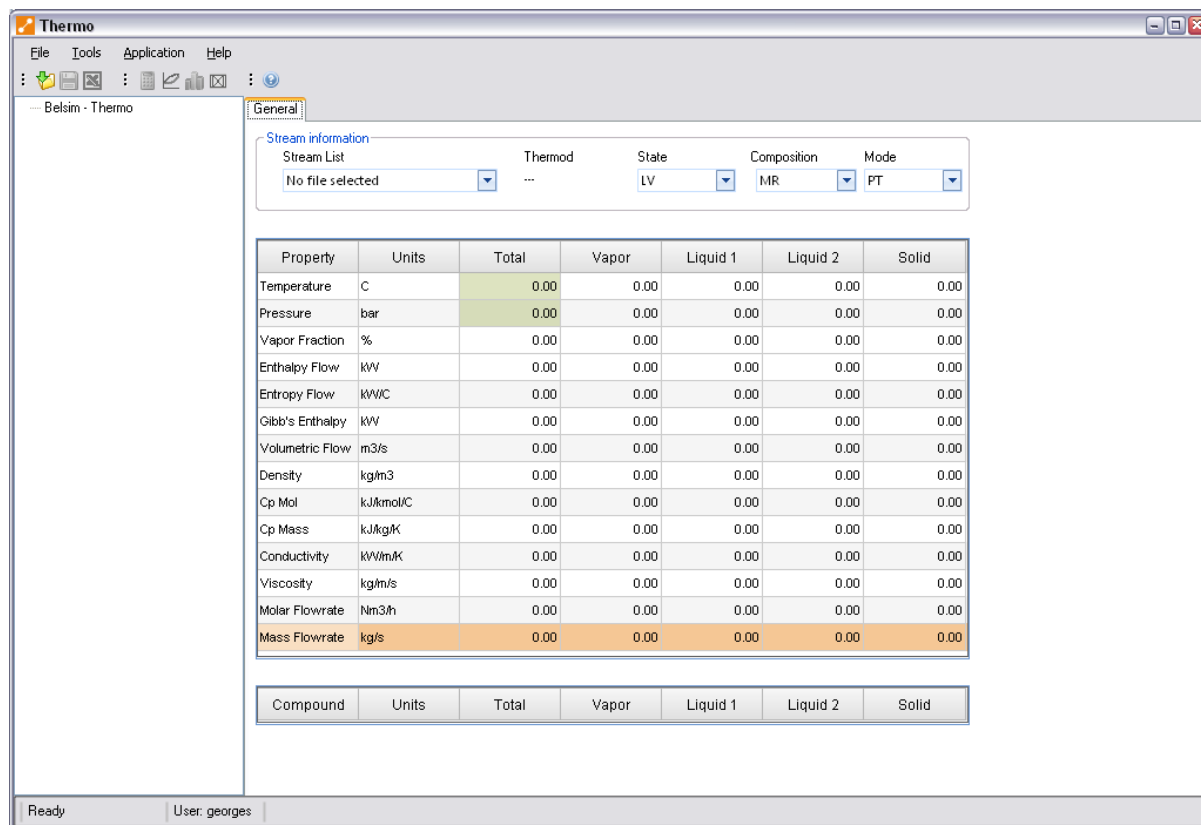


FIGURE 2 - THERMO EMPTY SCREEN

Thermo opens by default the model file loaded during the last use. If Thermo does not find the model file or if Thermo is opened for the first time, most functions are disabled until a model file is loaded.

2.2 OVERVIEW OF THERMO

Thermo window is divided in 5 main parts as illustrated in the picture below .

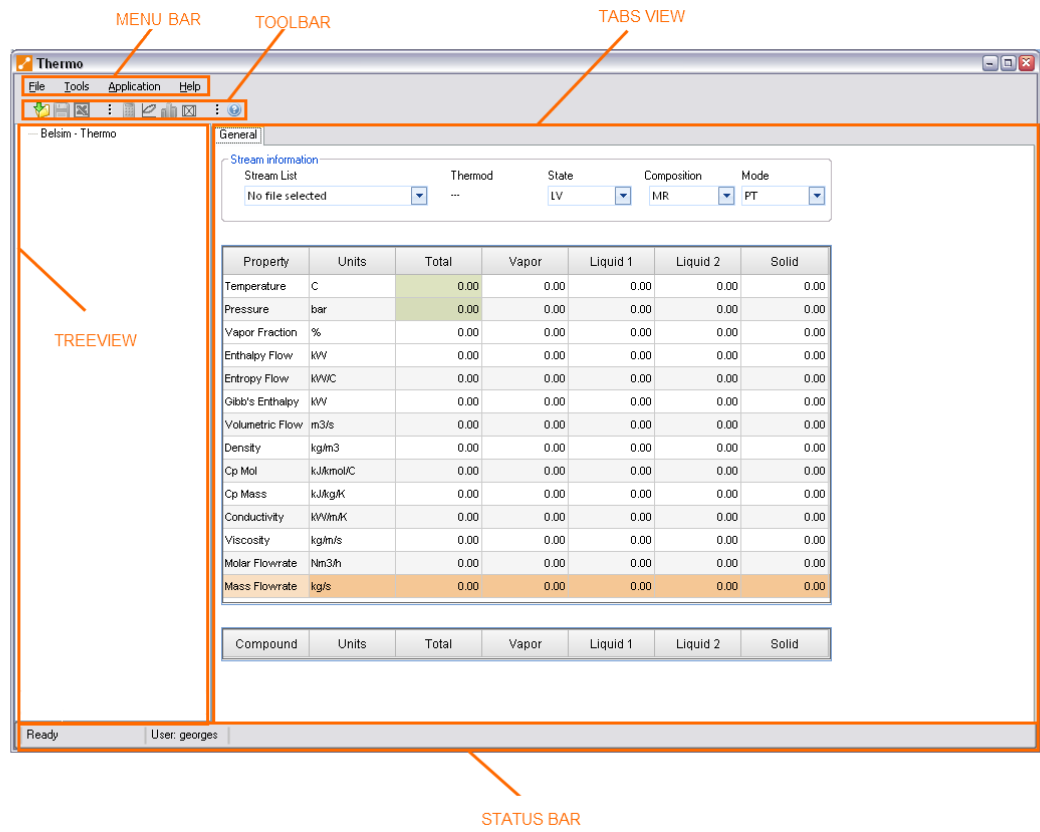


FIGURE 3 - THERMO WINDOW - OVERVIEW

2.3 MENU BAR

The Menu bar provides an access to all main functionalities of the program.

- **File**
 - use File |  Open to load an existing model file
 - use File |  Save model to save current stream properties into the model file
 - use File |  Export to Excel to export the displayed results in an Excel file
 - use File |  Exit to exit the program
- **Tools**
 - use Tools | Physical Units to define the physical units and the number format of properties to be displayed
 - use Tools | Save Results to save parameters and computed values in Thermo
 - use Tools | Load Results to load results that have been saved previously
- **Application**
 - use Application |  Calculate to compute thermodynamic properties of the selected stream under given computation mode
 - use Application |  White Equilibrium to perform the thermodynamic equilibrium of a stream using the WHITE method
 - use Application |  L/V Equilibrium to draw L/V equilibrium curves at constant pressure or temperature
 - use Application |  Draw Chart to draw customized charts. For a given stream, any of its properties can be plotted as a function of a state variable or a partial compound rate
- **Help**
 - use Help |  Content to open Thermo help
 - use Help | About to display general information about the application

2.4 TOOLBAR

The toolbar provides a quick access to main functions of the menu bar described above.

-  Open (File|Open)
-  Save model (File|Save model)
-  Export to Excel (File|Export to Excel)
-  Calculate (Application|Calculate)
-  L/V Equilibrium (Application|L/V equilibrium)
-  Draw Chart (Application|Draw Chart)
-  White Equilibrium (Application|White Equilibrium)
-  Help File (Help|Content)

2.5 TABS VIEW

The tabs view contains all the information about the selected stream. There are three major types of tab:

- General tab: it displays the selected stream information and enables to set the computation mode. It also displays results of the computation
- White equilibrium tab: it displays results of the White equilibrium computation
- Graph tab: it displays the created graphs. One graph is plotted per Tab

Tab can be closed by middle-clicking on it or by right-clicking to open the context menu and selecting the **Delete** button.

It is also possible to rename the tabs by the context menu, as shown below.

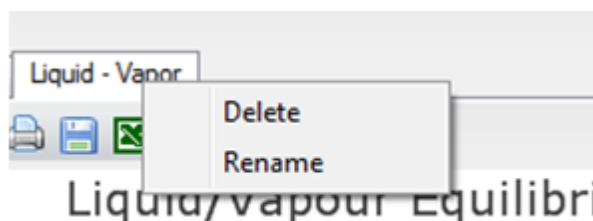


FIGURE 4 - CONTEXT MENU OF TABS

Note: the General tab cannot be closed or renamed.

2.6 STATUS BAR

The label on the left side of the status bar indicates the state of the last operation performed by Thermo.

The second label indicates the current Thermo user's name.

2.7 OPEN MODEL FILES

To open an existing model file, click on the  Open button in Menu|File or in the toolbar.

A dialog box appears for browsing the model file to be loaded.

Attention: model files created with the 4.6 or higher version of VALI can be opened by Thermo. Models created with an older version of VALI must be converted prior to using Thermo. The model file will be automatically converted once opened through the upgraded ValiModeller version

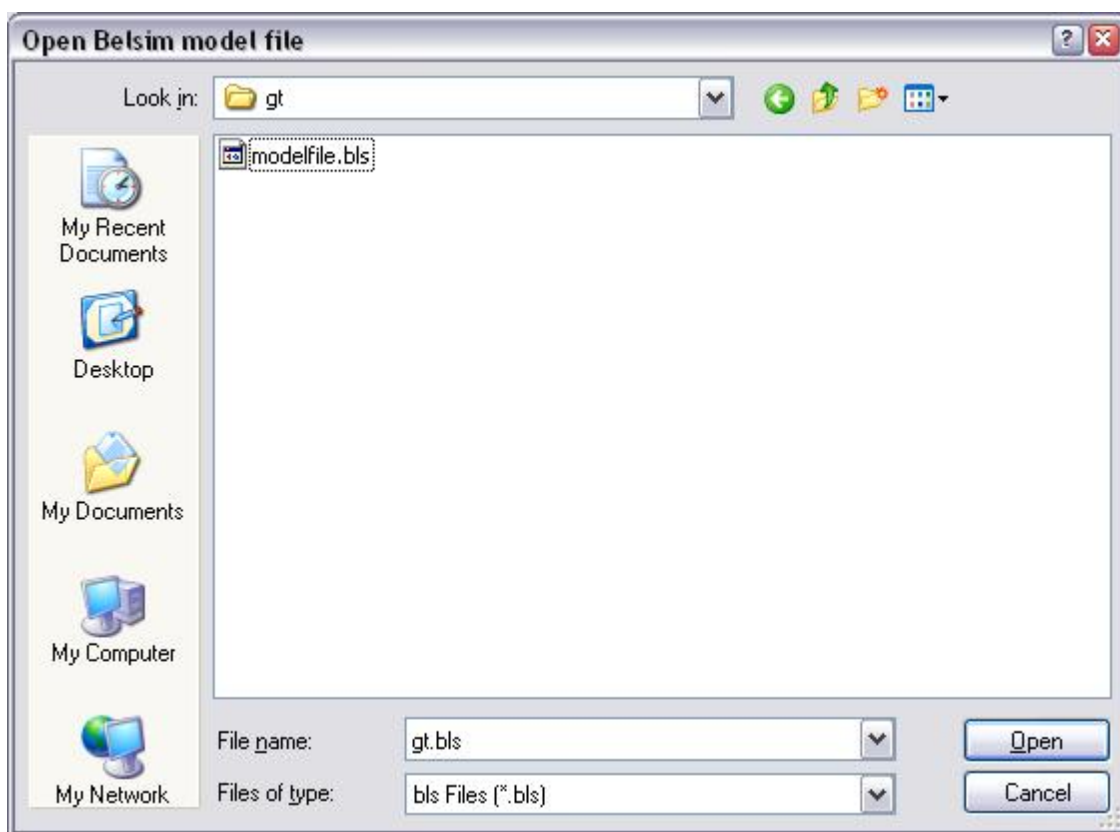


FIGURE 5 - OPENING MODEL FILE DIALOG BOX

When a valid model file is loaded, the properties of the first stream (alphabetically ordered) are automatically displayed in the General tab.

TREEVIEW

On the Thermo window left side, the Tree View shows the following nodes:

- the **Stream** node: it lists all the streams contained in the model file. It enables to load one stream by double-clicking on it
- the **Thermod** node: it displays the methods used in each Thermod and groups streams that use the same Thermod. It enables to load a stream by double-clicking on it
- the **Compounds** node: it groups streams by compounds. It enables to load a stream by double-clicking on it
- the **Charts** node: it groups created charts by stream. It enables to display a chart by double-clicking on it

Click on the  symbol to expand or on the  symbol to collapse the corresponding node of the Tree View.

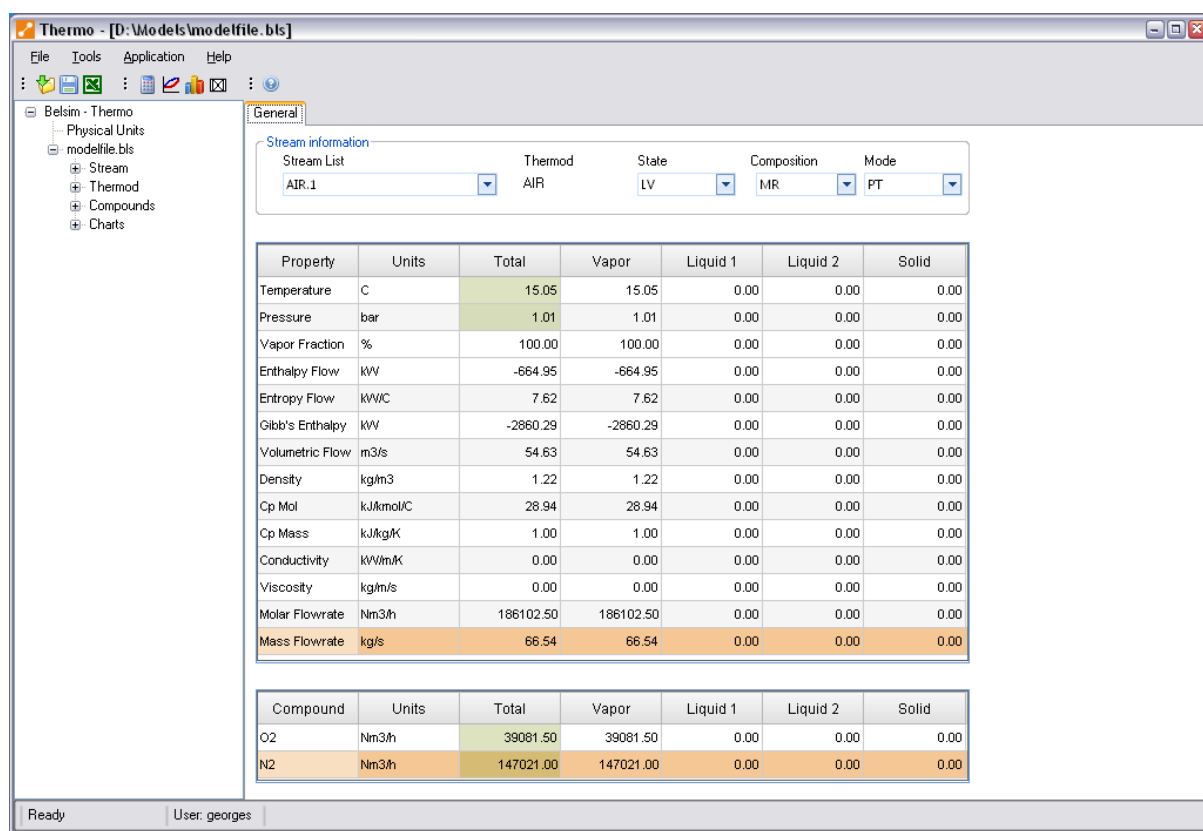


FIGURE 6 - THERMO WITH A MODEL FILE OPENED

2.8 THERMODYNAMIC PROPERTIES CALCULATION

Thermodynamic properties calculation can be performed once a model file has been correctly loaded.

The **Stream information** frame in the **General tab** displays the essential parameters used in the calculations.

Stream information

Stream List	Thermod	State	Composition	Mode
AIR.1	AIR	LV	MR	PT

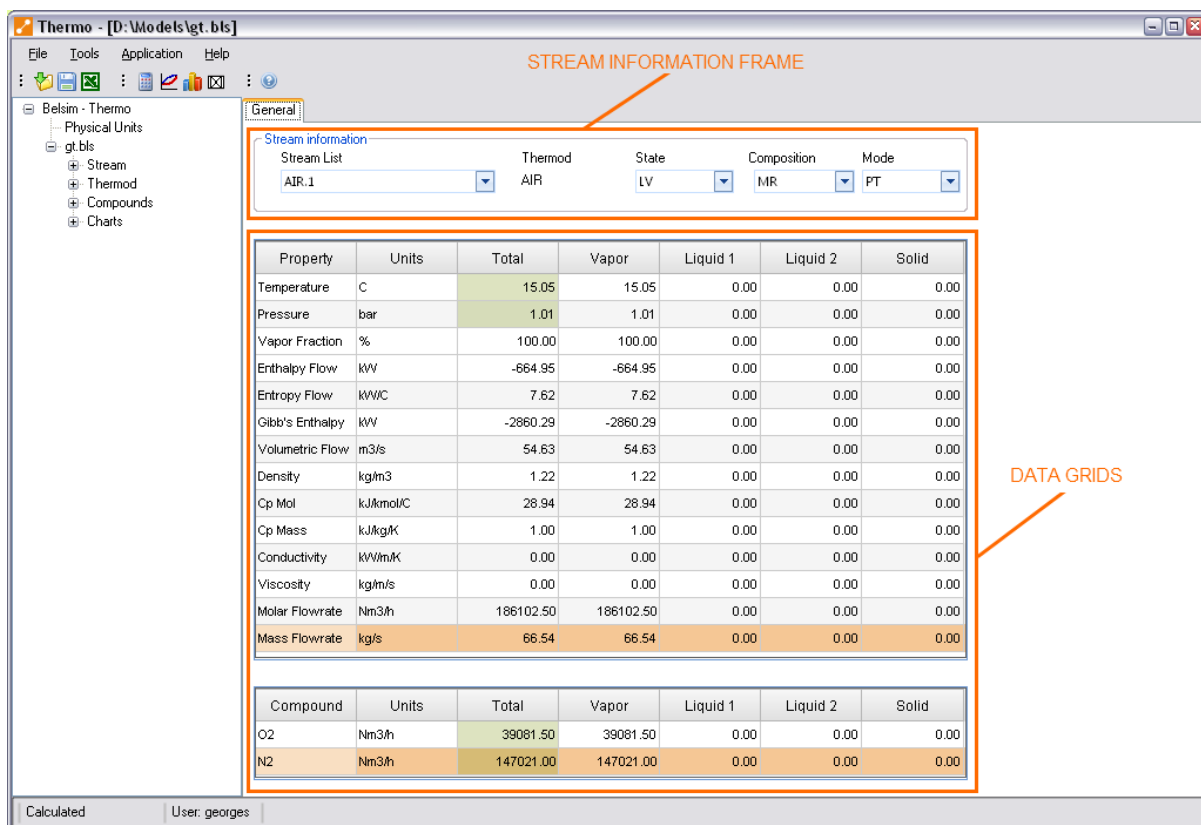
FIGURE 7 - STREAM INFORMATION FRAME

- **Stream list:** it shows the current stream selected. To select another stream, use the drop down list or the Tree View.
- **Thermod:** it indicates the Thermod of the selected stream. The Thermods are defined in the model file and cannot be modified through Thermo. Modifications of the Thermod have to be done via ValiModeller
- **State:** it indicates the state of the selected stream. To select another state, use the drop down list. The State field sets the phases that can be present in the selected stream:
 - L: Liquid phase only
 - V: Vapor phase only
 - LV: Liquid and Vapor phases
 - LL: two Liquid phases (if the model behavior enables demixing)
 - LLV: two Liquid phases and one Vapor phase
 - WL: Water phase and Liquid phase
 - WLV: Water, Liquid and Vapor phases
- **Composition:** it indicates the type of composition to be displayed. To select another composition, use the drop down list.
 - MR: Partial molar flow rate
 - WR: Partial weight flow rate
 - MF: Molar fraction
 - WF: Weight fraction
- **Mode:** it indicates the calculation mode of the selected stream. To select another mode, use the drop down list. The Mode field sets the two state variables used to determine the stream properties.
 - PT: Pressure – Temperature
 - PA: Pressure – Vapor fraction
 - TA: Temperature – Vapor fraction
 - PH: Pressure – Enthalpy
 - PS: Pressure – Entropy

The cells containing the values related to the selected calculation mode are highlighted in green. Double click on the green cells to modify the values of the state variables.

Once stream information and parameters of calculation are selected, click on the  Calculate button in the Menu/Application or in the toolbar to run the calculation. Results are displayed in the data grids.

Note: Thermo only takes into account the green cells to perform the calculations.



The screenshot shows the Thermo software window with the 'General' tab selected. The 'Stream information' section is highlighted with an orange box and labeled 'STREAM INFORMATION FRAME'. It contains dropdown menus for 'Stream List' (AIR.1), 'Thermod' (AIR), 'State' (LV), 'Composition' (MR), and 'Mode' (PT). Below this, two data grids are shown, both highlighted with orange boxes and labeled 'DATA GRIDS'. The first grid lists various properties (Temperature, Pressure, Vapor Fraction, etc.) and their values for different phases (Total, Vapor, Liquid 1, Liquid 2, Solid). The second grid lists compounds (O2, N2) and their flow rates (Nm3/h) for the same phases. The 'Total' column in both grids has green cells, indicating calculated values.

Property	Units	Total	Vapor	Liquid 1	Liquid 2	Solid
Temperature	C	15.05	15.05	0.00	0.00	0.00
Pressure	bar	1.01	1.01	0.00	0.00	0.00
Vapor Fraction	%	100.00	100.00	0.00	0.00	0.00
Enthalpy Flow	kW	-664.95	-664.95	0.00	0.00	0.00
Entropy Flow	kW/C	7.62	7.62	0.00	0.00	0.00
Gibb's Enthalpy	kW	-2860.29	-2860.29	0.00	0.00	0.00
Volumetric Flow	m3/s	54.63	54.63	0.00	0.00	0.00
Density	kg/m3	1.22	1.22	0.00	0.00	0.00
Cp Mol	kJ/kmol/C	28.94	28.94	0.00	0.00	0.00
Cp Mass	kJ/kg/K	1.00	1.00	0.00	0.00	0.00
Conductivity	W/m/K	0.00	0.00	0.00	0.00	0.00
Viscosity	kg/m/s	0.00	0.00	0.00	0.00	0.00
Molar Flowrate	Nm3/h	186102.50	186102.50	0.00	0.00	0.00
Mass Flowrate	kg/s	66.54	66.54	0.00	0.00	0.00

Compound	Units	Total	Vapor	Liquid 1	Liquid 2	Solid
O2	Nm3/h	39081.50	39081.50	0.00	0.00	0.00
N2	Nm3/h	147021.00	147021.00	0.00	0.00	0.00

FIGURE 8 - STREAM INFORMATION FRAME AND DATA GRIDS OF THE GENERAL TAB

DATA GRIDS

The data grid displays the stream properties and rates. These values are classified in five columns according to the phases of the selected stream. Two liquid phases can be represented if the model behavior allows demixing. The total phase represents the properties of the whole mixture.

By right-clicking on the data grids, the context menu is displayed.

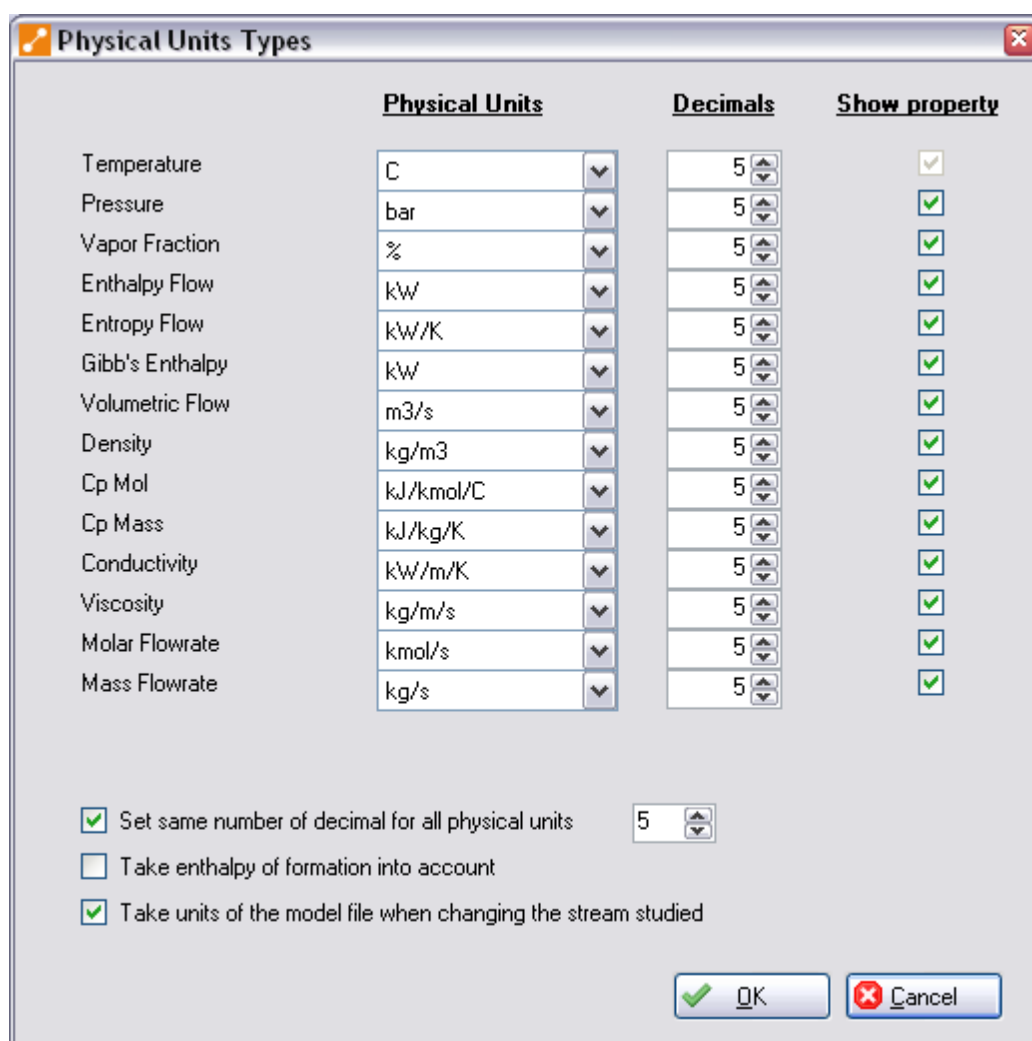
Property	Units	Total	Vapor
Temperature	°C	15.04920	15.04920
Pressure	bar	1.01167	1.01167
Vapor Fraction		0.00000	100.00000
Enthalpy Flow	kJ/s	-664.94574	-664.94574
Entropy Flow	kJ/s·K	7.61746	7.61746
Gibb's Enthalpy	kJ	-2860.29265	-2860.29265
Volumetric Flow	m ³ /s	54.62867	54.62867
Density	kg/m ³	1.21806	1.21806

FIGURE 9 - CONTEXT MENU OF THE DATA GRIDS

- Cut: cut the value of the selected cell
- Copy: copy the value of the selected cell into the clipboard
- Paste: paste the value of the clipboard in the selected cell
- Set as Total Column: fill the values of the Total column with the ones of the selected columns
- Show old: when activated, the results of the previous calculation are shown when the mouse quits the data grid

2.9 CHANGING PHYSICAL UNITS

To change the physical units, click on the Physical Units button in Menu/Tools or in the Tree View. The physical unit window is displayed and shows the current physical units used and their number format.



The dialog box titled "Physical Units Types" contains a table with three columns: "Physical Units", "Decimals", and "Show property". It lists 15 physical properties with their corresponding units and decimal settings. At the bottom, there are three checkboxes and a numeric input field for decimals, along with "OK" and "Cancel" buttons.

	Physical Units	Decimals	Show property
Temperature	C	5	☐
Pressure	bar	5	☒
Vapor Fraction	%	5	☒
Enthalpy Flow	kW	5	☒
Entropy Flow	kW/K	5	☒
Gibb's Enthalpy	kW	5	☒
Volumetric Flow	m3/s	5	☒
Density	kg/m3	5	☒
Cp Mol	kJ/kmol/C	5	☒
Cp Mass	kJ/kg/K	5	☒
Conductivity	kW/m/K	5	☒
Viscosity	kg/m/s	5	☒
Molar Flowrate	kmol/s	5	☒
Mass Flowrate	kg/s	5	☒

☒ Set same number of decimal for all physical units 5

☐ Take enthalpy of formation into account

☒ Take units of the model file when changing the stream studied

OK Cancel

FIGURE 10 - CHANGE PHYSICAL UNIT WINDOW

- to change the physical units, use the drop down list below the Physical Units label
- to change number format, use the drop down list below the Decimals label
- to hide/show a physical property in the data grids, use the drop down list below the Show property label

The three check boxes at the bottom of the window enable to respectively:

- set the same number of decimal for all elements of data grids at once
- take into account the enthalpy of formation for the enthalpy calculation
- automatically use the selected stream physical units as defined in the model file when selecting a stream in the General tab

2.10 SAVE RESULT IN THE MODEL FILE

To save stream properties calculated by Thermo into the currently loaded BLS model file, click on the  **Save model** button in Menu | Tools or in the toolbar.

A dialog box appears allowing you to select the stream of the model file into which the results are to be saved. The stream can be selected via the **Destination Stream** drop down list. The **Source Phase** drop down list enables to save the results of a specific phase. The **Source Stream** shows the stream results that are to be saved.

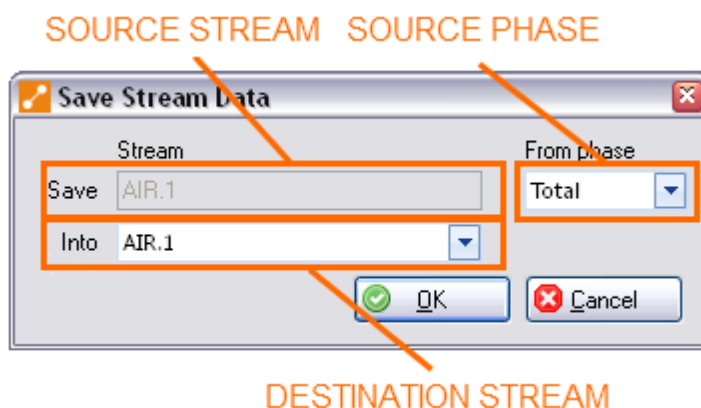


FIGURE 11 - SAVING PROMPT

Note: it is not possible to save the displayed results of a stream into another model file than the one currently loaded by Thermo.

2.11 EXPORT TO EXCEL

To export the displayed results of the data grids in an Excel file, click on the  **Export to Excel** button in Menu | File or in the toolbar.

A dialog box appears for browsing the directory and the name of the Excel file to be saved.

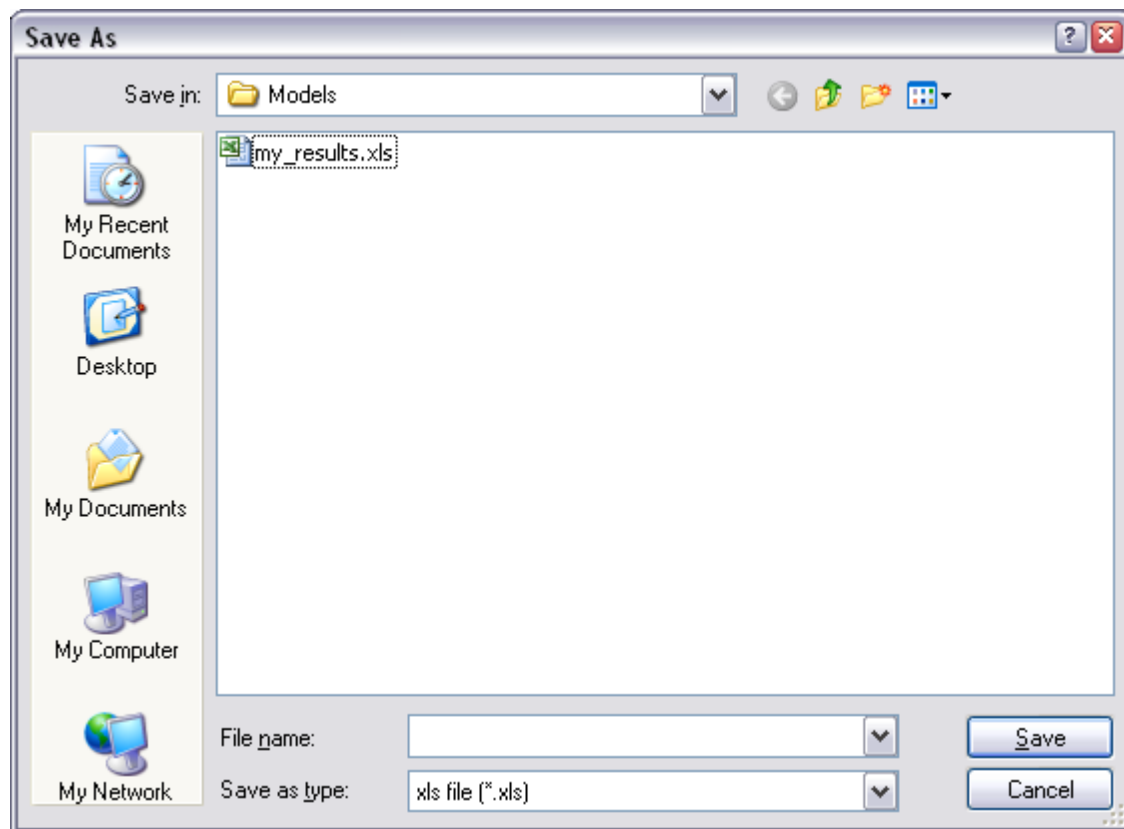
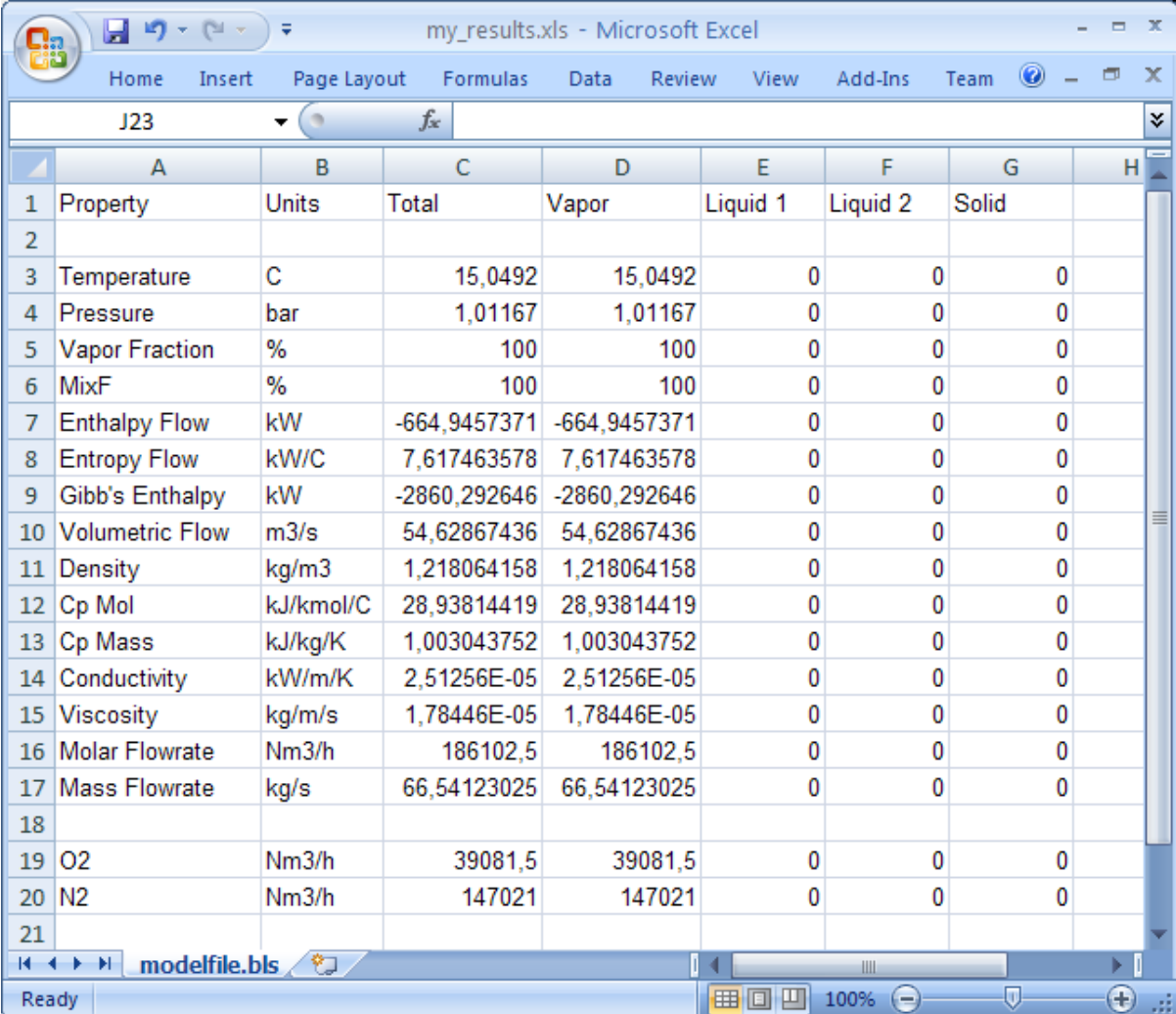


FIGURE 12 - EXPORT TO EXCEL DIALOG BOX

The structure of the data in the excel file is similar to the one of the data grids.



	A	B	C	D	E	F	G	H
1	Property	Units	Total	Vapor	Liquid 1	Liquid 2	Solid	
2								
3	Temperature	C	15,0492	15,0492	0	0	0	
4	Pressure	bar	1,01167	1,01167	0	0	0	
5	Vapor Fraction	%	100	100	0	0	0	
6	MixF	%	100	100	0	0	0	
7	Enthalpy Flow	kW	-664,9457371	-664,9457371	0	0	0	
8	Entropy Flow	kW/C	7,617463578	7,617463578	0	0	0	
9	Gibb's Enthalpy	kW	-2860,292646	-2860,292646	0	0	0	
10	Volumetric Flow	m3/s	54,62867436	54,62867436	0	0	0	
11	Density	kg/m3	1,218064158	1,218064158	0	0	0	
12	Cp Mol	kJ/kmol/C	28,93814419	28,93814419	0	0	0	
13	Cp Mass	kJ/kg/K	1,003043752	1,003043752	0	0	0	
14	Conductivity	kW/m/K	2,51256E-05	2,51256E-05	0	0	0	
15	Viscosity	kg/m/s	1,78446E-05	1,78446E-05	0	0	0	
16	Molar Flowrate	Nm3/h	186102,5	186102,5	0	0	0	
17	Mass Flowrate	kg/s	66,54123025	66,54123025	0	0	0	
18								
19	O2	Nm3/h	39081,5	39081,5	0	0	0	
20	N2	Nm3/h	147021	147021	0	0	0	
21								

FIGURE 13 - EXAMPLE OF RESULTS EXPORTED TO EXCEL

2.12 DRAWING LIQUID – VAPOR EQUILIBRIUM

This method calculates the Liquid – Vapor Equilibrium curves of a mixture as a function of one changing compound rate at constant pressure or constant temperature.

To draw Liquid-Vapor Equilibrium, click on the  **L/V Equilibrium** button in Menu|Application or in the toolbar. A dialog box prompts you to set the graph parameters.

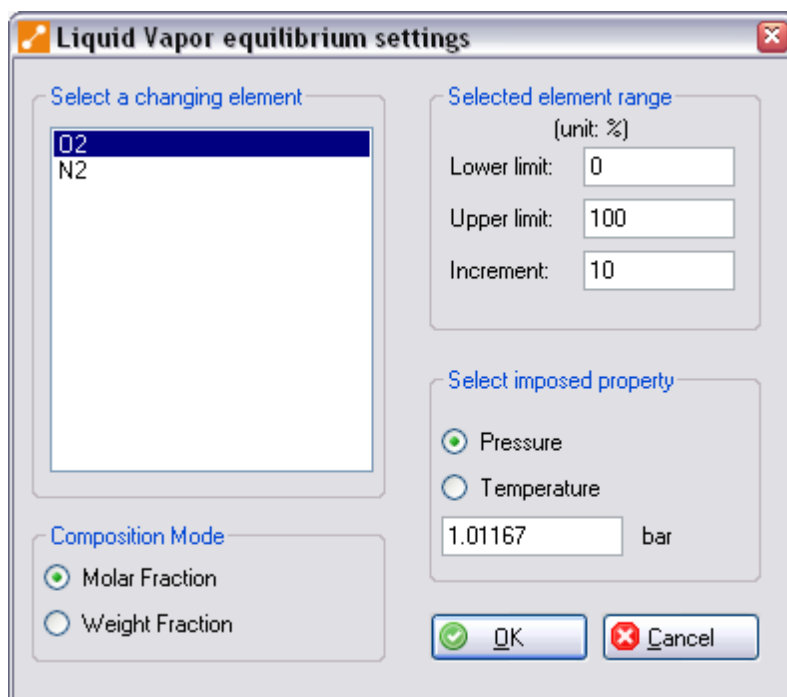


FIGURE 14 - LIQUID VAPOR EQUILIBRIUM SETTINGS

The parameters to be chosen are:

- **Select a changing element:** it selects the partial fraction flow rate to be plotted
- **Selected element range:** it sets the range and step of the changing element
- **Composition Mode:** it determines whether the selected element range is represented in molar or weight fraction
- **Select imposed property:** it enables to choose to either perform Liquid-Vapor equilibrium at fixed pressure or fixed temperature. A field enables to set the value of the fixed property

Once the parameters are set, click on the  **OK** button. The graph is plotted in a new tab. By default, the name of this tab is **Liquid – Vapor**.

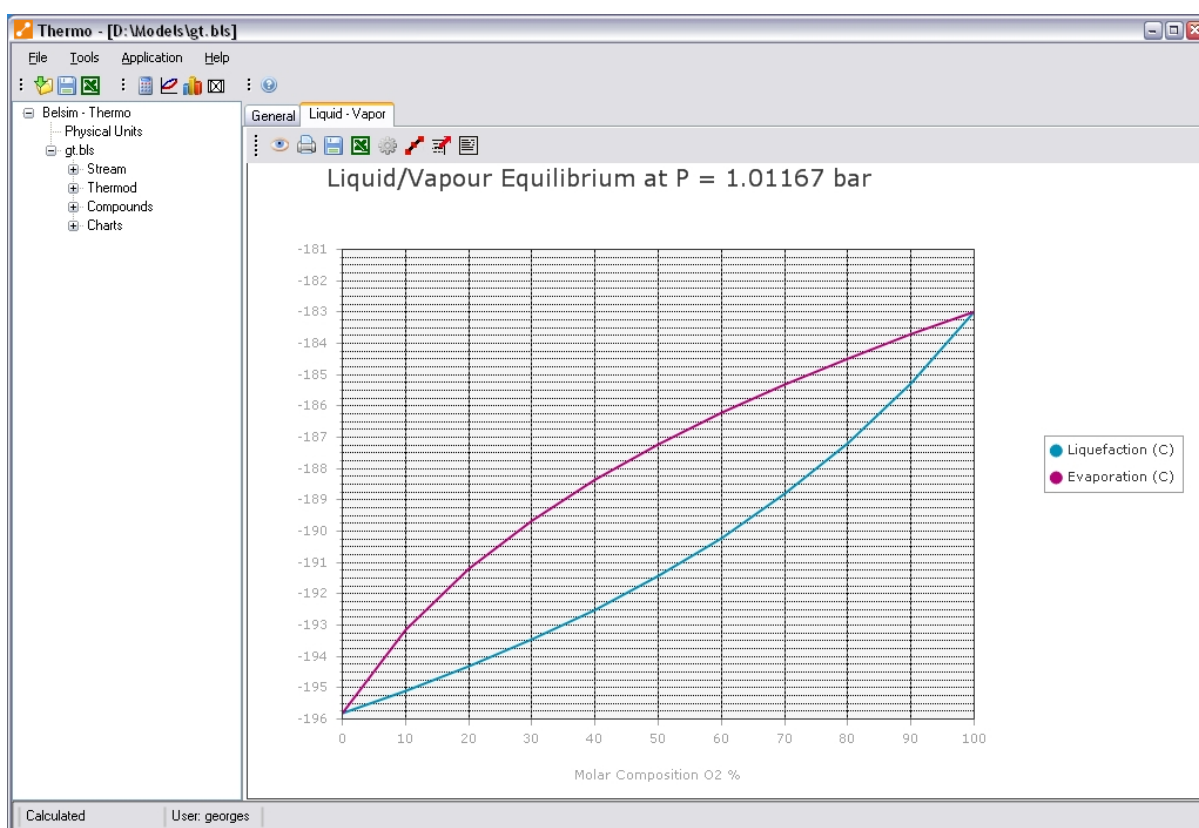


FIGURE 15 - AN EXAMPLE OF LIQUID VAPOR EQUILIBRIUM (AIR AT 1 ATM)

2.13 DRAW CUSTOMIZED CHART

The function Draw Chart aims at plotting the evolution of any properties in function of either a linearly changing state variable or a stream compound.

To draw a customized chart, click on the  **Draw Chart** button in the Menu|Application or in the toolbar. A dialog box prompts you to set the graph parameters.

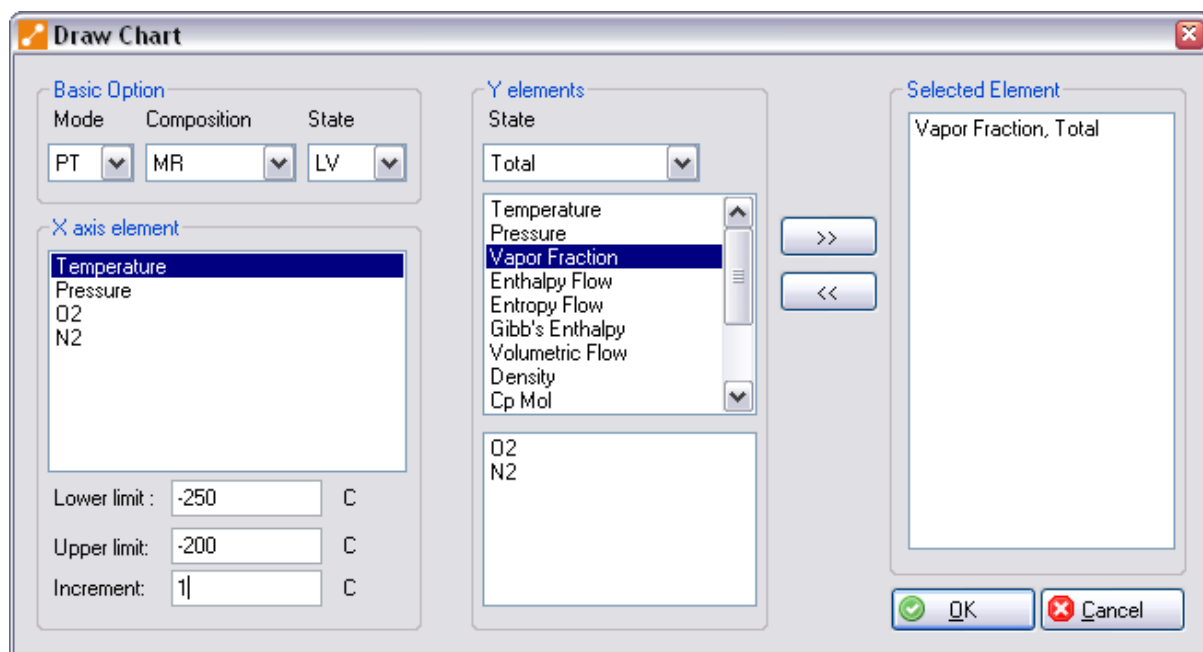
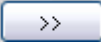
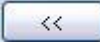


FIGURE 16 - DRAW CHART SETTINGS

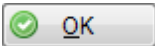
The parameters to be chosen are:

- **Basic Option** frame: it enables to change the calculation parameters as the **Mode**, the **Composition** and the **State**
- **X axis element** frame: It enables to choose
 - the element to be plotted on the horizontal axis
 - the variation range and step of the horizontal element

The X axis element can only be a state variable, a partial or total rates flow. Note that only state variables taken into account for calculation are displayed. Change the **Mode** to display other state variables

- **Y elements** frame: it enables to choose any property or any of the partial rates to be plotted on the vertical axis. To select an element to be plotted:
 - select the element in the **Y elements** frame (the first box contains the thermodynamics properties and the second the compounds of the mixture)
 - select the stream phase in the list box below the **State** label to get the values of the selected property related to the specific phase
 - click on the  Add button
- **Selected Element** frame: It shows the selected elements to be plotted in the Y axis. To remove one of these elements from the list, select it and click on the  Remove button

Important note: the X axis element is chosen depending on the selected mode while the other state variables of the stream (including composition) are determined through the data grids of the general tab. Please go to the general tab and double-click on the variable cell to change its value.

Once the graph parameters are set, click the  OK button. The graph is plotted in a new tab. By default, the name of this tab is **Chart**.

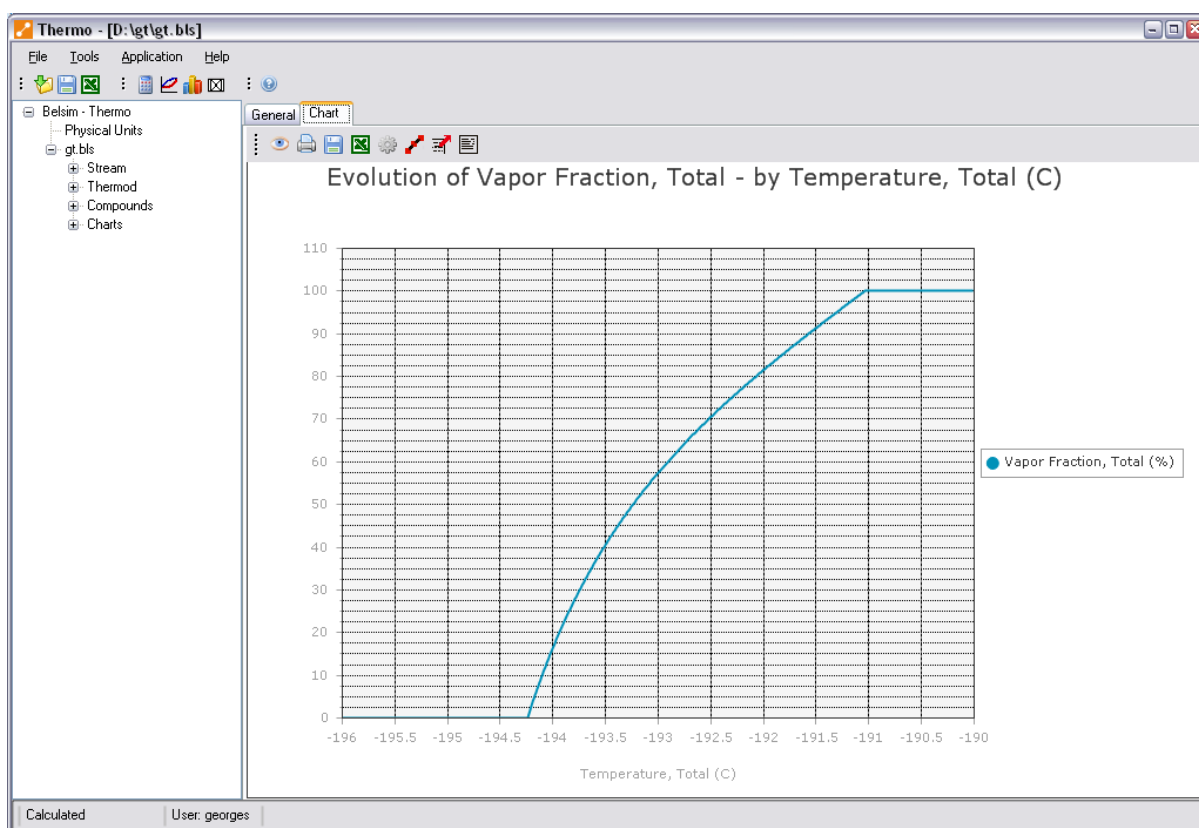


FIGURE 17 - AN EXAMPLE OF CUSTOMIZED CHART (VAPOR FRACTION OF THE AIR IN FUNCTION OF THE TEMPERATURE)

2.14 GRAPH TAB

A **Graph tab** is displayed when

- a Liquid-Vapor Equilibrium graph is created
- a customized graph is created

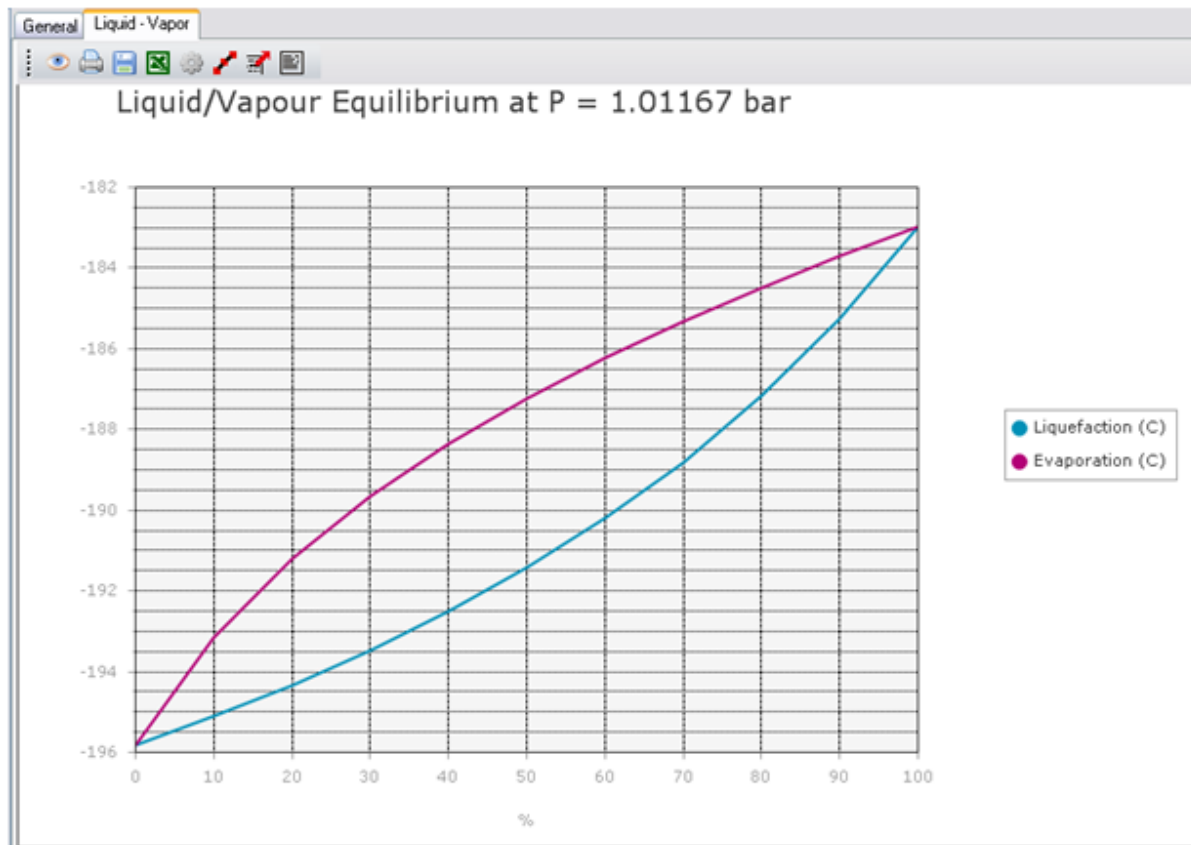


FIGURE 18 - EXAMPLE OF GRAPH TAB

By right-clicking on the graph, the context menu is displayed.

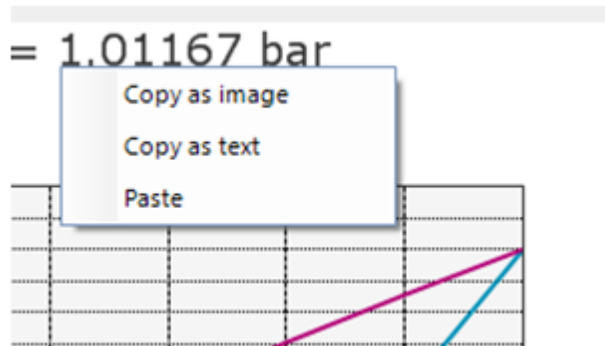


FIGURE 19 – CONTEXT MENU OF THE GRAPH

- use Copy as image to Copy the chart as an image in the clipboard
- use Copy as text to Copy the chart data in an Excel format in the clipboard
- use Paste to merge a graph that has been previously copied as text with the one currently displayed

The **toolbar** on the top of the Graph tab provides access to specific functions

- use  Show Values to show the values of each point of the graph
- use  Print to print the graph
- use  Save as to export the graph in a new image file
- use  Export to Excel to export the graph data in a new Excel file
- use  Preferences to open the preference window
- use  Show Points to show or hide the point of the graph
- use  Open in new Window to open the graph in a new window
- use  Show Legend to show or hide the legend of the graph. The legend gives information about the parameters used to plot the graph such as the composition, the changing property, the stream or the model file used.

CHART LAYOUT

The **preference** window is composed of four tabs enabling to customize the graph layout.

The **General tab** enables setting

- the plot color
- the background color
- the title (text, font, size, color and background color)

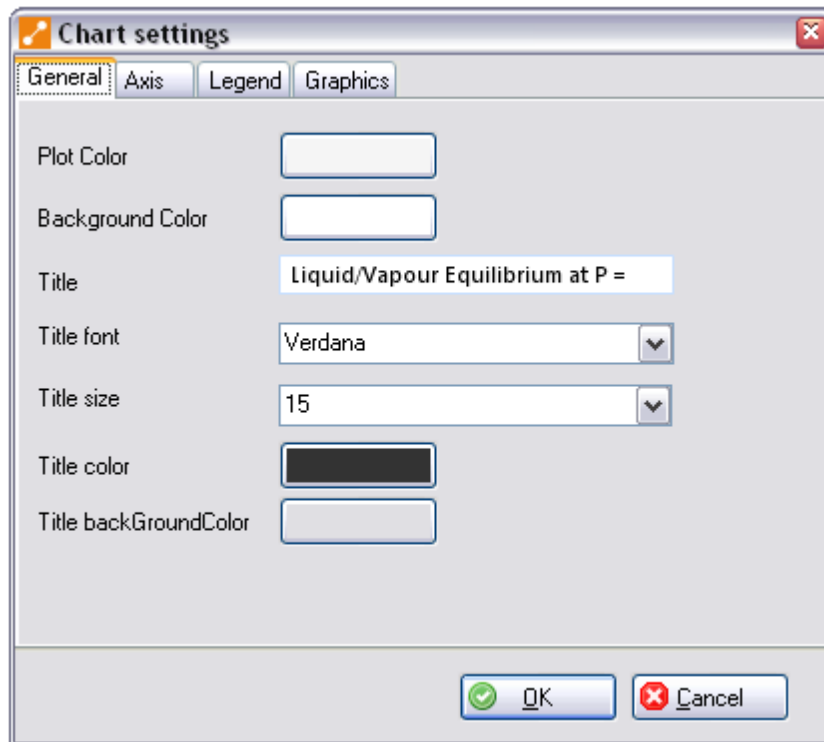


FIGURE 20 – CHART SETTINGS - GENERAL TAB

The **Axis tab** enables setting

- the title of the axis
- to shown/hide the grid
- to display axis in logarithmic mode
- to set the area of the axis

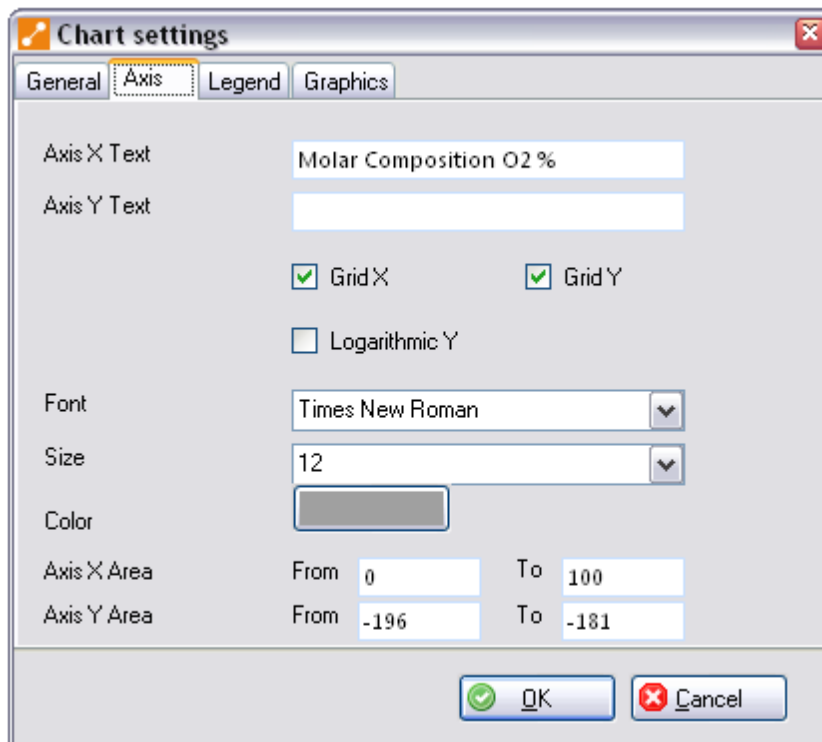


FIGURE 21 – CHART SETTINGS – AXIS TAB

The **Legend tab** enables to see the legend of the graph and to set the information to be shown.

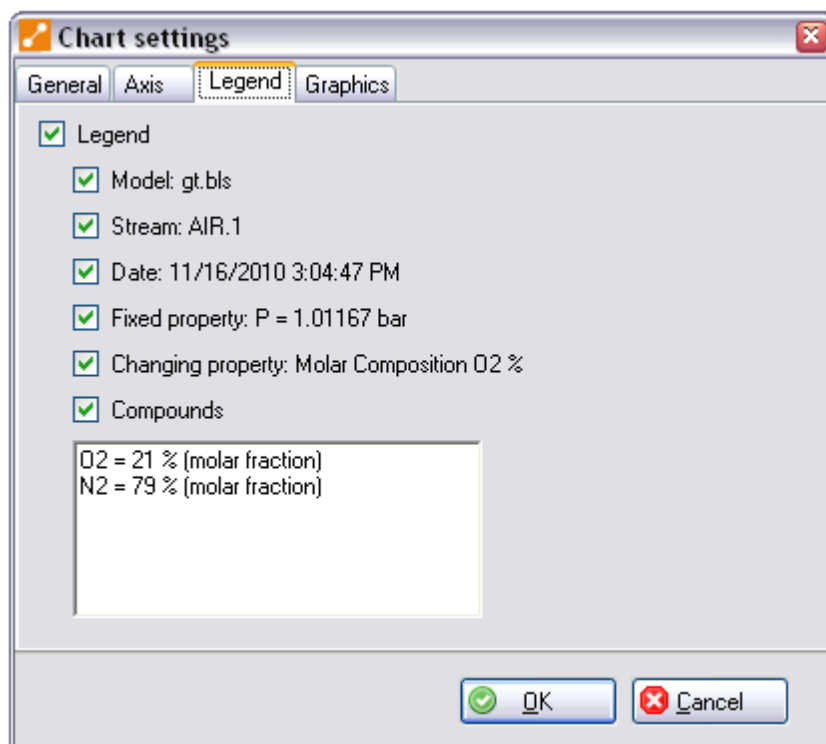


FIGURE 22 – CHART SETTINGS – GRAPHICS TAB

The **Graphics tab** enables to change the layout of the curves

- the line color
- the markers of the calculated points
- the line style
- show or hide the selected curve on the graph

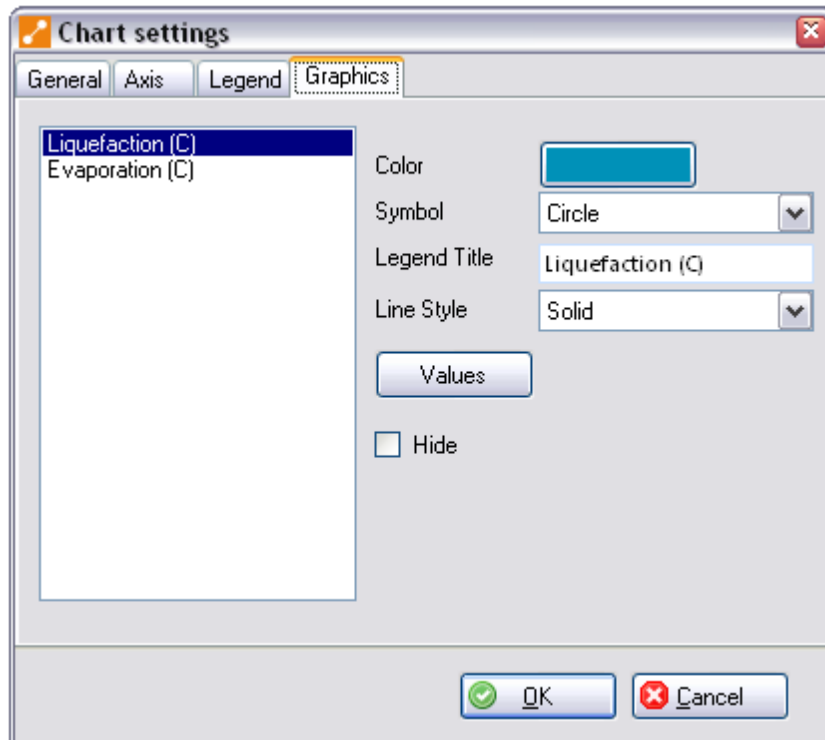
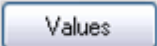
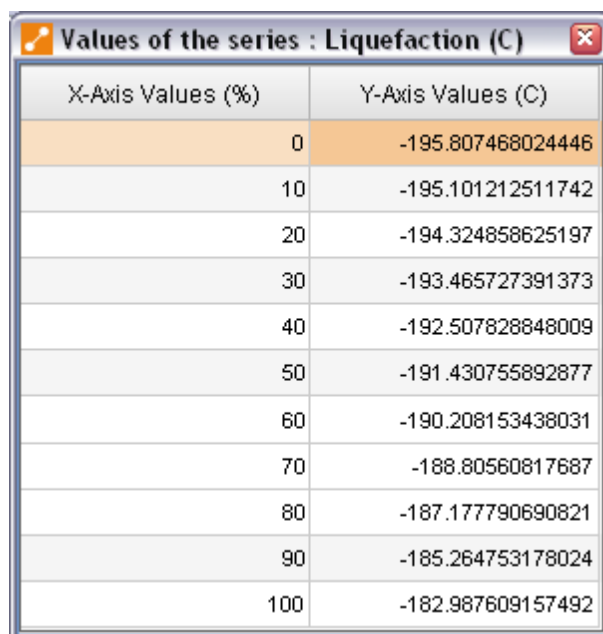


FIGURE 23 – CHART SETTINGS

By clicking on the  **Values** button, a grid is displayed showing the values of the calculated points of the selected curve.



X-Axis Values (%)	Y-Axis Values (C)
0	-195.807468024446
10	-195.101212511742
20	-194.324858625197
30	-193.465727391373
40	-192.507828848009
50	-191.430755892877
60	-190.208153438031
70	-188.80560817687
80	-187.177790690821
90	-185.264753178024
100	-182.987609157492

FIGURE 24 - AN EXAMPLE OF CALCULATED POINTS OF A CURVE

2.15 WHITE EQUILIBRIUM

The White Equilibrium function performs a thermodynamic equilibrium of a stream using the WHITE method. It takes into account all the possible chemical reactions in the mixture and minimizes the free Gibb's Energy in a range of temperature or enthalpy. WHITE method involves chemical reaction and therefore it uses the formation enthalpy for enthalpy display.

To start the equilibrium computation, click on the ☒ **White Equilibrium** button in Menu Application or in the toolbar.

A dialog box prompts you to set the parameters.

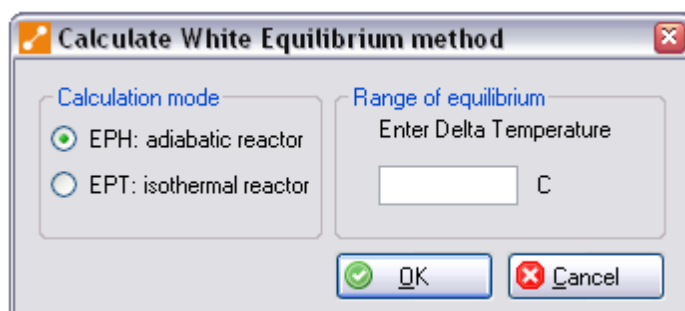


FIGURE 25 - WHITE EQUILIBRIUM DIALOG BOX

Two **calculation modes** are available:

- EPH (adiabatic reactor): equilibrium calculated at the enthalpy and pressure set in the General tab
- EPT (isothermal reactor): equilibrium calculated at the temperature and the pressure set in the General tab

The **Delta Temperature (Enthalpy)** represents the deviation from the equilibrium. It means that the Gibb's Enthalpy is minimized at a temperature in a range $T \pm \Delta T$ (enthalpy of $H \pm \Delta H$) to find the equilibrium. The mixture properties are shown for the temperature T (enthalpy H).

Note: White equilibrium is designated for VAPOUR phases only and calculations will be done assuming a **vapor phase** mixture.

The **White Equilibrium** tab is displayed once the white equilibrium has been calculated.

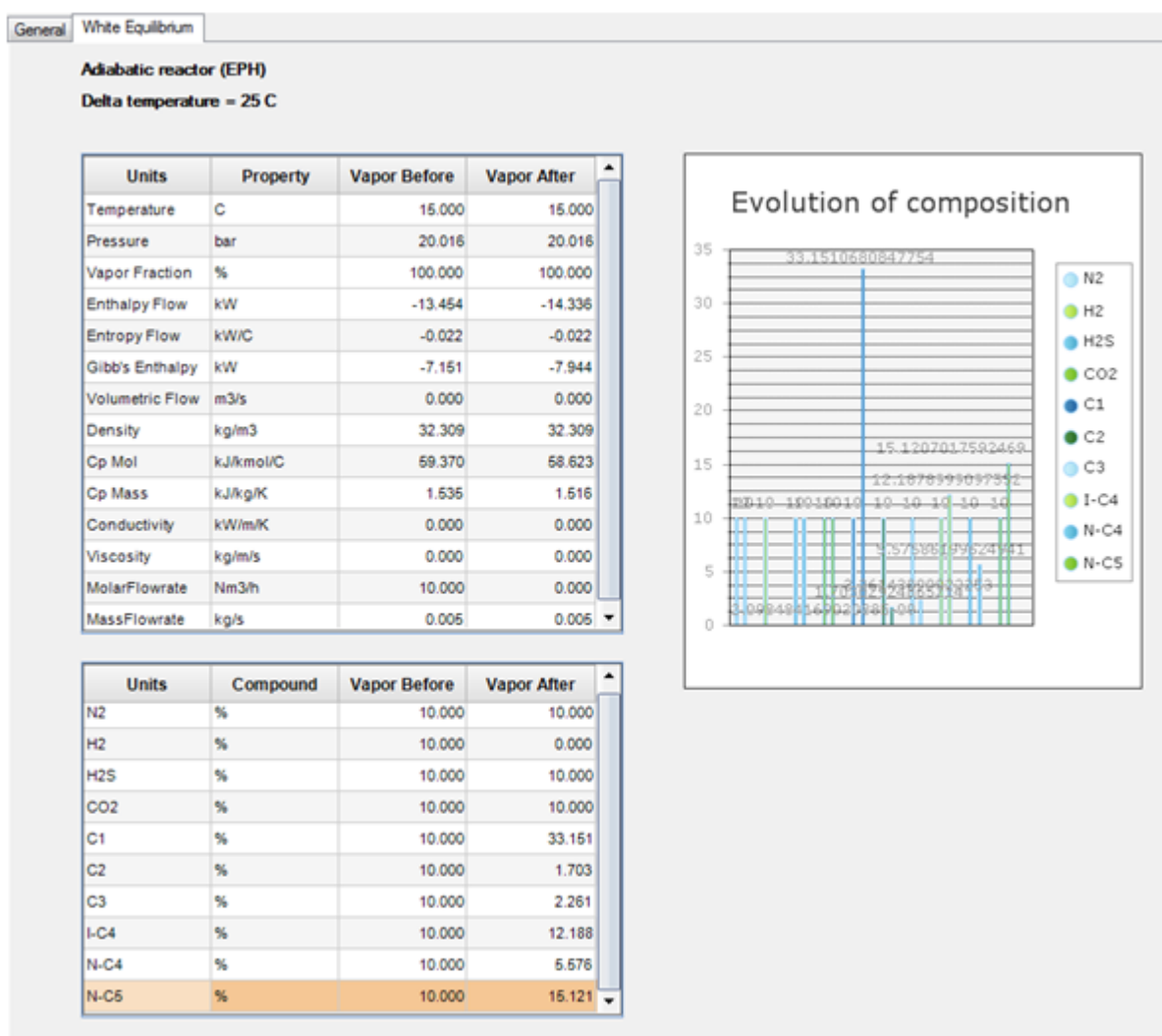


FIGURE 26 - WHITE EQUILIBRIUM TAB

The two grids on the left side display the properties before and after the runs for the vapor phase. The graph on the right side gives an overview of the variation of the composition.

2.16 SAVING, LOADING AND COMPARING RESULTS

SAVING RESULTS

It is possible to save the displayed stream properties in **Thermo** in order to reload them later on or to compare them with other computations.

The stream, the model file and the calculation parameters used are saved together with the results.

To save results:

- click on the Save results button in the Menu | Tools
- enter a name for the saved results in the dialog box
- click the OK button



FIGURE 27: SAVE RESULTS DIALOG BOX

LOADING RESULTS

Once the results are saved, they can be reloaded by clicking on the **Load Results** button in Menu | Tools. It opens a dialog box with all the previously saved results. Select the saved results by the check box on the right side and click on the load button. The model file is automatically loaded and the saved results are displayed in data grids of the **General tab**.

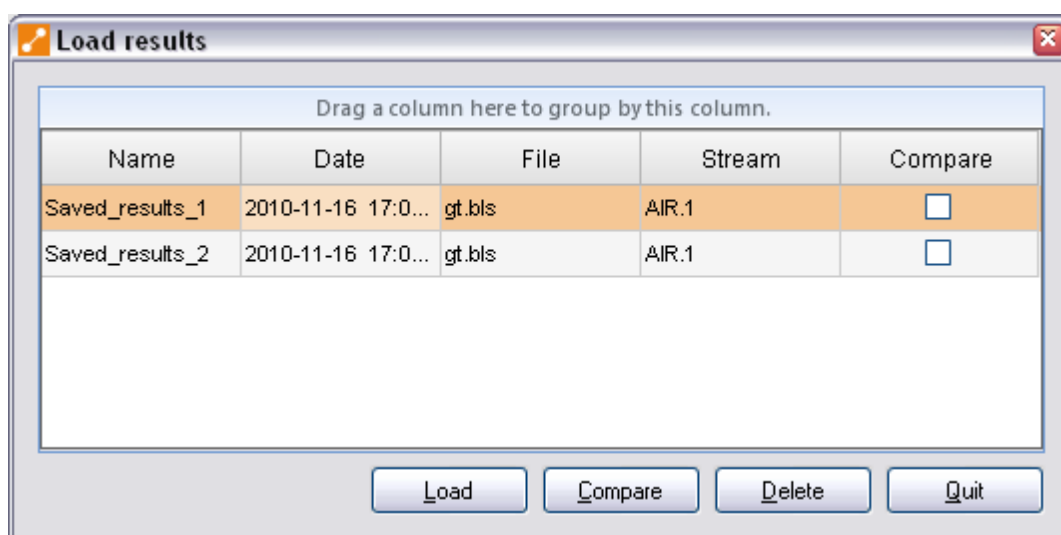


FIGURE 28 - LOAD RESULTS DIALOG BOX

COMPARE RESULTS

The Load Result dialog box also enables to select several results to be compared. Select the saved results by the check boxes on the right side and click on the **Compare** button. Under the condition that selected streams have the same **Thermod**, a new tab is created with the selected streams.

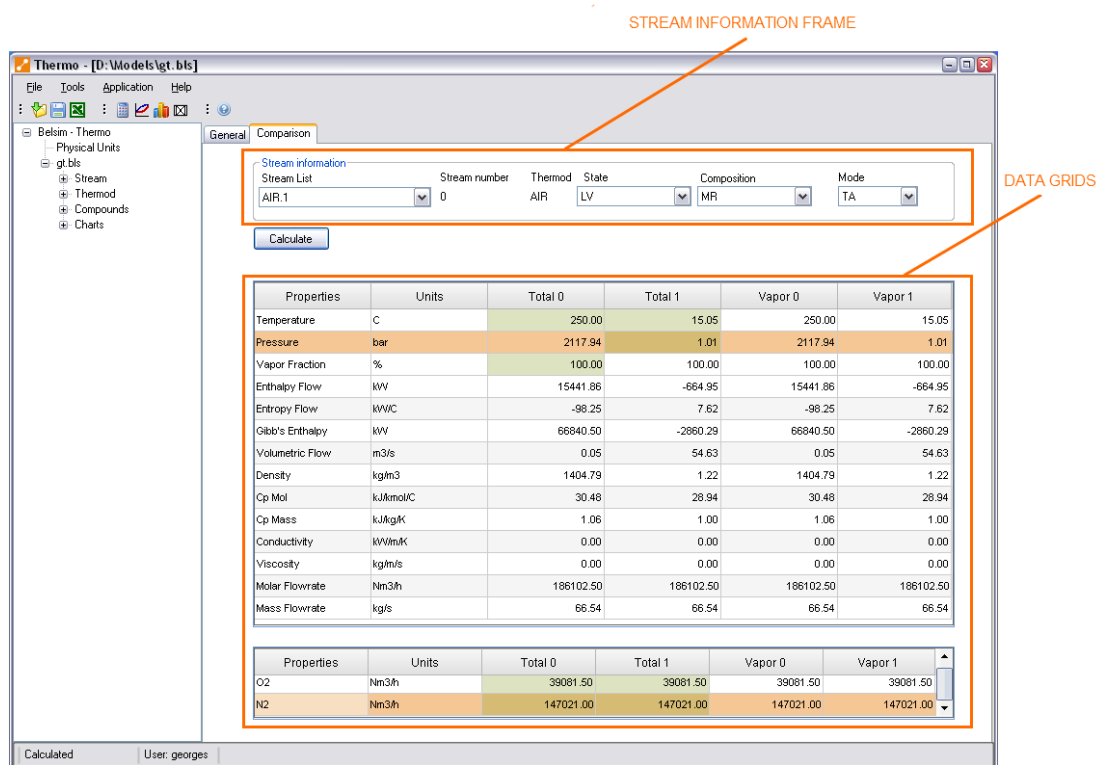


FIGURE 29 - COMPARISON TAB

Stream information frame is similar to the one of the **General tab**. It contains:

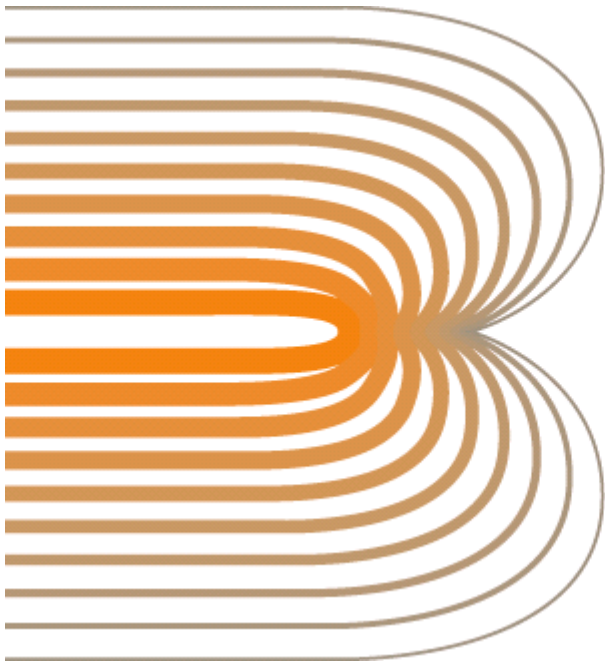
- **Stream List:** it contains the selected streams to be compared
- **Stream number:** it indicates the number of the stream. This number is used as a reference to the stream in the data grids
- **State:** it indicates the state of the selected stream
- **Composition:** it indicates the composition of the selected stream
- **Mode:** it indicates the calculation mode of the selected stream

The **data grids** contain the results of the streams comparison. Columns that correspond to those phases that have all rates equal to zero are not shown to ensure a better readability of the results. Column header names are composed of the stream phase followed by a number (the Stream number) that refers to the streams loaded. For example, if the stream number is 0, the **Vapor 0** column designates thermodynamic properties of the stream in the Vapor phase.

Through this window, thermodynamic properties calculation on several streams can be done at once.

For each stream of the **Stream List**, set the stream parameters and computation mode in the **Stream information** frame. The parameters can be different for each stream.

Enter the state variables and partial compounds rate (green cells) with desired values and click the **Calculate** button located under the Stream information frame to launch the computation. The results are displayed in data grids.



belsim

Your performance wizard