

XSOL Mapping

User Guide



DOCUMENTATION VERSION: 3.1.7

Copyright *XSOL Limited* 1999-2006. All rights reserved.

Table of contents

1	Introduction	13
2	How do I use this help?	14
3	Business Process Management (BPM)	15
3.1	Definitions and examples	16
3.1.1	Enterprise	16
3.1.2	Trading Entities	16
3.1.3	Business Processes	17
3.1.4	Roles	17
3.1.5	Groups	17
3.1.6	Processes	18
3.1.7	Stages	18
3.1.8	Tasks	18
3.1.9	Process Flow	19
3.1.10	Tables	20
3.1.11	Resources	20
3.1.12	Resource Extensions	20
3.1.13	Control Groups	21
3.1.14	Relationships	21
4	How do I use XSOL Mapping?	22
4.1	Logging on	22
4.1.1	Logging off	22
4.2	Using XSOL Mapping in a multiple-user environment	23
4.3	Using the XSOL health check	24
4.4	General functions	25
4.4.1	Windows	25
4.4.2	Navigation	26
4.4.3	Customising designer buttons	27
4.4.4	Properties Editor	27
4.4.5	Drop-down menus	28

4.4.6	Resizing items	29
4.4.7	Sorting columns	29
4.4.8	Undoing and redoing changes	30
4.4.9	Copying and pasting	30
4.4.10	Groups	32
	Changing a Group	33
4.4.11	Deleting items	33
4.5	Designing a model	34
4.5.1	Saving a model	35
4.5.2	Packaging a model	36
	Unpacking a model	37
4.5.3	Loading a model	37
4.5.4	Restoring a model	37
4.5.5	Upgrading a model	38
4.5.6	Removing model version information	38
4.6	Designing an Enterprise	39
4.6.1	Changing an Enterprise	39
4.6.2	Deleting an Enterprise	39
4.6.3	Designing a Trading Entity	39
	Changing a Trading Entity	40
4.6.4	Using Logical Printers	40
	Changing a Logical Printer	41
4.6.5	Using Document Locations	41
	Adding a limit to a Document Location	42
	Changing a Document Location	42
4.6.6	Using System Values	43
4.6.7	Defining User Notes	45
4.7	Designing Business Processes	48
4.7.1	Using Processes	48
	Defining Variables that map to an initiated Process	50
	Changing a Process	50
4.7.2	Using Companion Processes	50
	Sharing Variables from a Companion Process to a Process	52
	Adding Variables to support external initiation	52
	Changing a Companion Process	52
	Using Initiation Rules	53
	Changing an Initiation Rule	54
4.7.3	Using Stages	54
	Changing a Stage	55
4.7.4	Using Escalation Levels	56
	Using exception flow	57

	Changing an Escalation Level	58
4.7.5	Designing a To Do List display	58
	Changing a To Do List display	60
4.7.6	Using Tasks	60
	Mapping a Variable from one Process to another	66
	Adding an attachment to a Send Email Task	67
	Selecting a document in XSOL Automation	67
	Using a Microsoft Office template with a Task	69
	Using the Script Task	71
	Changing a Task	73
	Deleting a Task	73
4.7.7	Changing the display of Stage/Task nodes	74
4.7.8	Using Process Summaries	74
	Changing a Process Summary	76
	Using Process Summary Filters	77
	Changing a Process Summary Filter	79
4.8	Designing Process Flow	80
4.8.1	Adding Process Flow nodes	80
	Adding a Start/Stop node	82
	Adding a Stage/Task node	82
	Adding a Joint node	83
	Adding an All/Any node	83
	Adding a Condition node	85
	Adding a Decision node	85
	Adding a Fork node	87
	Adding an Annotation node	88
	Adding a Group node	88
4.8.2	Using Swim Lanes	89
4.8.3	Opening Task Process Flow	90
4.8.4	Automatically creating Process Flow diagrams	90
4.8.5	Changing a Process Flow icon	91
4.8.6	Changing the default icon for a Process Flow node	92
4.8.7	Aligning Process Flow nodes	93
4.8.8	Linking Process Flow nodes	94
4.8.9	Changing Process Flow node properties	95
4.8.10	Validating Process Flow links	95
4.8.11	Selecting Process Flow nodes	96
4.8.12	Viewing Process Flow	96
4.8.13	Viewing available items in Task Process Flow	97
4.8.14	Filtering Process Flow by Role	98
4.8.15	Printing a Process Flow diagram	99

4.8.16	Printing options for documentation generation	100
4.8.17	Using reference captions	100
4.9	Designing Roles	102
4.9.1	Assigning a Role	102
4.9.2	Filtering by Role	103
4.9.3	Changing a Role	103
4.9.4	Deleting a Role	103
4.10	Designing Worksheets	105
4.10.1	Changing a Worksheet	106
4.10.2	Using widgets	107
	Widget properties	108
	Changing a widget	109
4.10.3	Aligning widgets on a Worksheet	110
4.10.4	Setting the Tab order on a Worksheet	111
4.11	Designing Tables	112
4.11.1	Using Resources	112
	Changing a Resource	113
4.11.2	Using Resource Extensions	113
	Changing a Resource Extension	114
4.11.3	Using Control Groups	115
	Using model defined Control Groups	116
	Adding or editing model defined Control Group entries	116
	Deleting a model defined Control Group entry	117
	Changing a Control Group	117
4.11.4	Using Relationships	117
	Changing a Relationship	119
4.11.5	Using datafields	119
	Changing a datafield	123
	Input Mask syntax	123
	Using a Display Mask	125
4.11.6	Using Secondary Indices	126
	Changing a Secondary Index	127
4.12	Designing Variables	128
4.12.1	Using Process Variables	128
	Selecting a user in XSOL Automation	132
	Changing a Variable	133
4.12.2	Using Reference Variables	133
	Changing a Reference Variable	134
	Using Business Rules with Reference Variables	134
	Using Datafield Variables	135
	Changing a Datafield Variable	135

4.13	Designing Lookup Lists	136
4.13.1	Changing a Lookup List	137
4.14	Designing Expressions	138
4.14.1	Assigning an Expression	140
4.14.2	Changing an Expression	140
4.14.3	Using Parameters	141
	Changing a Parameter	142
4.14.4	Checking an Expression's syntax	142
4.14.5	Certifying an Expression	143
4.14.6	Certifying all Expressions	144
4.15	Using User Notes	145
4.15.1	Using Word User Notes	146
4.15.2	Using the "Normal" style with User Notes	147
4.15.3	Changing a User Note	147
4.16	Validation	148
4.16.1	Validating the model	148
4.16.2	Validating specific items	149
4.17	Where Used	150
4.18	Generating documentation	151
4.18.1	Generating documentation outputs	151
4.18.2	Using documentation filters	152
	How selecting Roles changes your generated documentation	153
4.18.3	Using documentation layouts	154
4.18.4	Using HTML custom sections	157
4.18.5	Using HTML documentation templates	159
4.18.6	Using styles with the HTML output	160
4.18.7	Setting up HTML Help generation	161
4.18.8	Emailing generated HTML documentation	161
4.18.9	Linking to generated HTML documentation	162
4.18.10	Using Word 2003 custom sections	162
4.18.11	Using Word 2003 documentation templates	164
4.18.12	Using styles with the Microsoft Word output	165
4.18.13	Printing tips	167
4.19	Options	169
4.19.1	General - options	169
4.19.2	New Model Defaults - options	170
4.19.3	Highlighting - options	171
4.19.4	Properties Editor - options	172
4.19.5	Icons - options	173
	Displaying Task icons	173
	Adding additional icons	173

Using Task icon associations	174
4.19.6 Documentation - options	175
4.19.7 Predefined Filters - options	176
4.19.8 Editor Options - options	176
4.19.9 Editor Colours - options	177
4.19.10 Default Icons - options	178
4.20 Keyboard shortcuts	180
5 Registration details	190
6 End-User License Agreement	191
7 Glossary	193
8 Appendix	197
8.1 Python modifications	197
8.1.1 Naming conventions	197
8.1.2 Basic types	197
8.1.3 Non-standard Python syntax	198
8.1.4 Reserved keywords	198
8.1.5 Global functions	200
average ()	200
coalesce ()	200
count ()	201
countBlank ()	201
hasNoValue ()	201
hasValue ()	201
max ()	201
min ()	202
sum ()	202
toString ()	202
8.1.6 Binary operations	202
8.1.7 Date/Time operations	203
Date ()	203
DateTime ()	204
Days ()	206
DaysDiff ()	206
Hours ()	206
HoursDiff ()	207
Minutes ()	207
MinutesDiff ()	207
Months ()	208

	Seconds ()	208
	SecondsDiff ()	208
	Time ()	208
	WeekDay ()	209
	WeekDayRange ()	210
	Weeks ()	211
	WeeksDiff ()	211
	Years ()	211
8.1.8	Document operations	211
	Document ()	211
	DocumentLocation ()	212
8.1.9	Email operations	212
8.1.10	File operations	213
	appendDirectory ()	213
	deleteFile ()	213
	exists ()	213
	FileName ()	213
	forceDirectories ()	214
	move ()	214
	pathList ()	215
	removeDirectories ()	215
8.1.11	Notational operations	215
8.1.12	Number operations	215
	FixedPoint ()	216
	float ()	217
	int ()	217
8.1.13	Percent operations	218
	Percent ()	219
8.1.14	Phone operations	220
8.1.15	Printer operations	220
8.1.16	Reference Variable operations	221
8.1.17	Script Task operations	221
	AddFileToDocument ()	221
	getTaskDataXml ()	221
8.1.18	Stream operations	224
	FileStream ()	224
	MemoryStream ()	226
	Stream ()	226
	StringStream ()	227
8.1.19	User operations	228
8.1.20	Useful techniques	228

8.2	Command Line Parameters	230
-----	-----------------------------------	-----

List of figures

3.1	Relationship of Business Process constructs	15
3.2	Business Model Hierarchy constructs	16
3.3	Process Flow example	19
4.1	Logging on	22
4.2	XSOL Mapping window description	25
4.3	Ordering the designer list	27
4.4	Using a menu	28
4.5	Resizing a Process Flow node	29
4.6	Sorting a columns	30
4.7	Copying and pasting a Resource	31
4.8	Resolving dependencies with a pasted Process	31
4.9	Automatically resolving dependencies when copying and pasting	32
4.10	Deleting a Stage	33
4.11	Saving a model	36
4.12	Packaging a model	36
4.13	Loading a model	37
4.14	Choosing the correct backup file	38
4.15	Designing an Enterprise	39
4.16	Adding a Trading Entity	40
4.17	Adding a Logical Printer	41
4.18	Changing a Logical Printer	41
4.19	Adding a Document Location	42
4.20	Adding a limit to a Document Location	42
4.21	Changing a Document Location	43
4.22	Defining a User Note	47
4.23	Designing Business Processes	48
4.24	Adding a Process	49
4.25	Defining Variables that map to an initiated Process	50
4.26	Adding a Companion Process	51
4.27	Sharing Variables from a Companion Process to a Process	52
4.28	Adding a Variable to support external Companion Process initiation	52
4.29	Adding an Initiation Rule	53

4.30 Adding a Stage	55
4.31 Setting the timeout required to activate the next Escalation Level	57
4.32 Using exception flow	58
4.33 Changing an Escalation Level	58
4.34 Designing a To Do List display	59
4.35 Changing the sort order of To Do List Variables	60
4.36 Changing a To Do List display	60
4.37 Adding a Task	66
4.38 Automatically mapping Variables from one Process to another	66
4.39 Adding an attachment to a Send Email Task	67
4.40 Setting predefined filters for a Document Variable	69
4.41 Mapping bookmarks to Variables	71
4.42 Adding Variables to a Script Task	71
4.43 Changing the display of Stage/Task nodes	74
4.44 Adding a Process Summary with a Stage drill-down	76
4.45 Changing a Process Summary	77
4.46 Setting the properties of a Process Summary Filter	78
4.47 Designing Process Flow	80
4.48 Process Flow diagram	81
4.49 Merging two separate flows	83
4.50 Repeat loop	83
4.51 All Stages must be completed for an All node	84
4.52 Only one Stage must be completed for an Any node	84
4.53 A Condition node	85
4.54 A Decision node	86
4.55 Outcomes for a Decision Table	87
4.56 A Fork node	87
4.57 Adding a Group node	89
4.58 Using Swim Lanes	90
4.59 Opening Task Process Flow	90
4.60 Automatically creating a Process Flow diagram	91
4.61 Changing a Process Flow icon	92
4.62 Changing a Process Flow icon	93
4.63 Aligning the bottom of selected nodes	94
4.64 Linking Process Flow nodes	94
4.65 Changing a Process Flow node's properties	95
4.66 Validating Stage Process Flow links	95
4.67 Selecting multiple Process Flow nodes	96
4.68 Closing the Expansion Panel	97
4.69 Creating a new Initiate process Task	98
4.70 Filtering a Process Flow diagram by Role	99

4.71 Printing a Process Flow diagram	99
4.72 Setting printing options for Process Flow diagrams	100
4.73 Using reference captions	101
4.74 Adding a Role	102
4.75 Assigning a Role	103
4.76 Filtering by Role	103
4.77 Opening the Design Worksheet Layout window	106
4.78 Changing a Worksheet	107
4.79 Adding a widget	107
4.80 Changing a widget	109
4.81 Aligning selected widgets horizontally	110
4.82 Setting the Tab order on a Worksheet	111
4.83 Adding a Resource	113
4.84 Changing a Resource	113
4.85 Adding a Resource Extension	114
4.86 Changing a Resource Extension	115
4.87 Adding a Control Group	115
4.88 Adding a Control Group entry	116
4.89 Changing a Control Group	117
4.90 Adding a Relationship	119
4.91 Changing a Relationship	119
4.92 Adding a datafield	123
4.93 Changing a datafield	123
4.94 Creating a Display Mask	126
4.95 Adding a Secondary Index	127
4.96 Changing a Secondary Index	127
4.97 Adding a Variable	132
4.98 Creating a Variable and Business Rule that allows you to select a user in XSOL Automation	132
4.99 Changing a Variable	133
4.100 Adding a Reference Variable	134
4.101 Adding a Business Rule to a Reference Variable	134
4.102 Adding a Datafield Variable	135
4.103 Adding a Lookup List	136
4.104 Changing a Lookup List	137
4.105 Functions list	139
4.106 Operators list	139
4.107 Adding a Macro Expression	140
4.108 Assigning an Expression	140
4.109 Changing an Expression	141
4.110 Adding a Parameter	141
4.111 No errors found	142

4.112Invalid syntax	142
4.113Certifying an Expression	143
4.114Adding a Designer User Note	146
4.115Validating the model	148
4.116Validating the model	149
4.117Checking where an item is used	150
4.118	152
4.119Editing a documentation filter	153
4.120Editing a documentation layout	157
4.121Inserting an HTML custom section	158
4.122Garamond applied to all paragraph tags	161
4.123Adding a custom section in Word 2003	164
4.124Changing a Word 2003 documentation template	165
4.125Printing a large diagram in landscape mode	168
4.126Printing a large diagram in portrait mode	168
4.127Setting General options	169
4.128Setting New Model Defaults options	170
4.129Setting highlighting options	172
4.130Setting options for the Properties Editor	173
4.131Hiding a specific Task icon	173
4.132Adding a new Task icon	174
4.133Adding a Task icon association	175
4.134Changing the documentation layout directory	175
4.135Adding a predefined filter	176
4.136Setting Editor options	177
4.137Setting Editor Colour options	178
4.138Setting Default Icons options	179
5.1 Checking registration details	190

1 Introduction

Welcome to *XSOL Mapping*'s reference manual. This document is designed as a reference to assist you to use *XSOL Mapping*.

If you have a support request and purchased *XSOL Mapping* through an authorized reseller, please contact your reseller for help.

If you have a support request and purchased *XSOL Mapping* directly from XSOL, please contact XSOL on +64 9 309 6135 (8 a.m. - 5:30 p.m., Monday to Friday) or help@xsol.com.

XSOL Mapping is a BPM (Business Process Management) tool. It allows you to:

- Model business processes for any business
- Design Process Flow diagrams
- Design data tables and datafields
- Generate documentation that includes all model information

2 How do I use this help?

This manual is designed to help you find the information you require as quickly as possible. If you are unsure how to do something or you have a problem, try the following approaches:

- **Table of contents** - The Table of contents allows you to scan through the topics that are available in a logical order. It is worth taking a few minutes and reading the Contents to familiarize yourself with the help.
- **Index** - The Index lists key words and where they appear in the help. Use the Index when you know the type of thing you are looking for but are unsure where the information might be.

Note, Important, and Warning graphics

Text that is particularly useful or important for some reason appears in the help in the following manner:



This text is a general note.



This text is an important note.



This text is an urgent warning.

3 Business Process Management (BPM)

XSOL Mapping is a Business Process Management (BPM) application that enables the design, analysis, and optimisation of Business Processes. It does this by:

- Separating the Process logic from the applications that run them
- Managing relationships among Process participants
- Integrating internal and external Process resources

This section describes the relationship of a business process construct with other constructs and the implication of any particular combination. The following diagram illustrates the relationships between Business Process constructs:

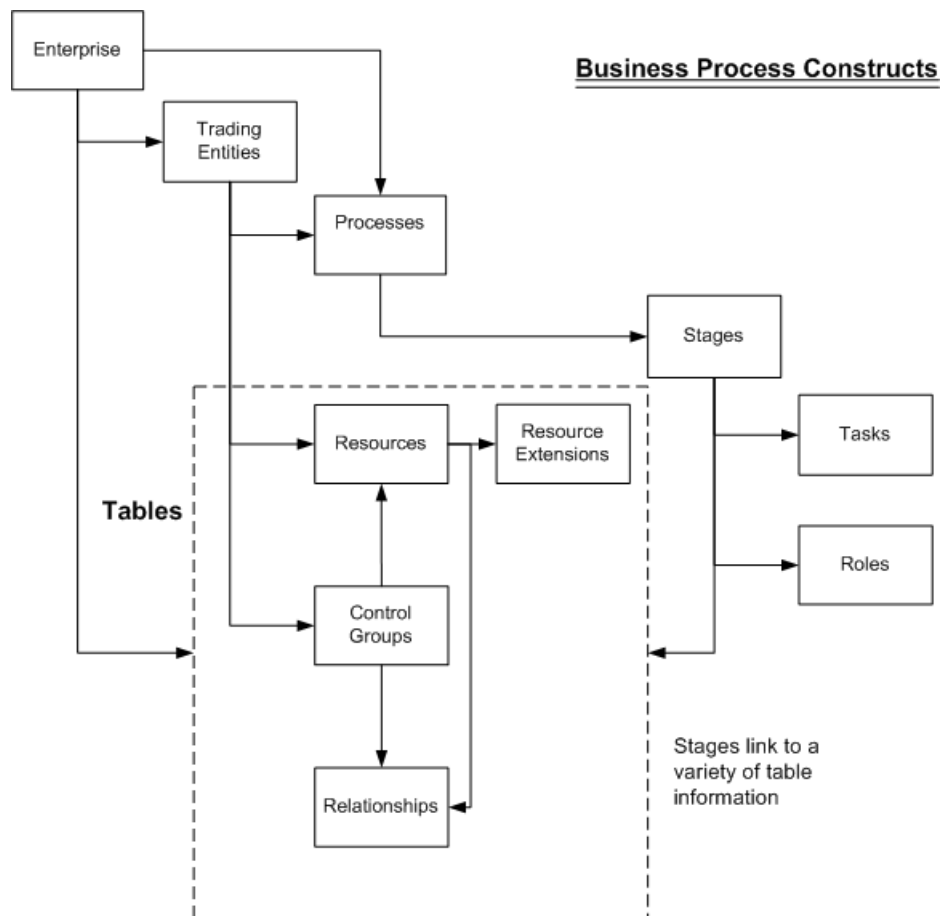


Figure 3.1: Relationship of Business Process constructs

The XSOL Business Model Hierarchy

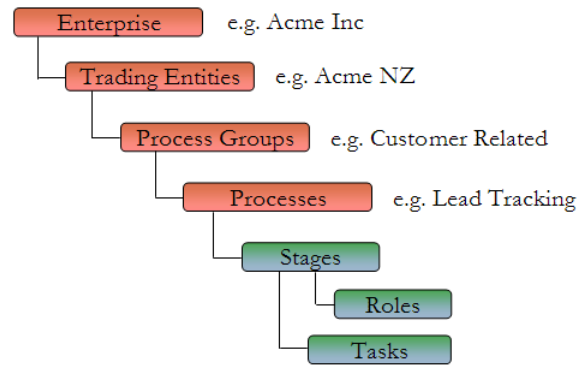


Figure 3.2: Business Model Hierarchy constructs

This top-down "picture" of the corporation is "packaged" as a two-dimensional model - a hierarchy of static business components and a horizontal time-based processing flow. This indicates the sequence in which the data is entered or altered and so determines the value of subsequent data.

This model forms the basis for ESL (Enterprise System Logic) - XSOL's predefined model of business activity that enables companies to define their business systems in the way the company actually works in real-life.

Please refer to the following sections for full definitions of each component of ESL (Enterprise System Logic).

3.1 Definitions and examples

This section defines each Business Process construct used in *XSOL Mapping* and what associations it has, along with examples.

3.1.1 Enterprise

Definition:

The Enterprise is the top node of the hierarchy. An Enterprise is a container for the Business Processes of one or more Trading Entities. The Enterprise has the same capabilities as a Trading Entity, thus Roles, Reference Data, and Processes can all be defined.

Example:

Acme Holdings is the parent company of Acme Distribution (a retail store). In this model Acme Holdings is the Enterprise and Acme Distribution is a Trading Entity.

Relationship with other elements:

- **Trading Entities** - Separate companies or divisions of an Enterprise. Refer to page 16 for more information.

3.1.2 Trading Entities

Definition:

A discrete entity within the Enterprise through which business is conducted; each Trading Entity will have its own Roles, Reference Data, Processes, and Arithmetic data aggregation defined.

Example:

Acme Distribution is a retail store within Acme Holdings.

Relationship with other elements:

- **Roles** - Roles can be added or changed at the Trading Entity level. Roles can be assigned at the Stage level.
- **Tasks** - Tasks can be added or changed at the Trading Entity level. Tasks can be assigned at the Stage level.
- **Data elements** - Data elements (Tables) are defined at the Trading Entity level.

3.1.3 Business Processes

Definition:

Business Processes describes how an entire business, or portion of a business, is run in business terms. This consists of what actions are required for a particular outcome, when an action needs to occur, and who performs that action.

3.1.4 Roles

Definition:

The class of user to whom a Stage is assigned who is normally associated with a functional business area; for example, either broad - Accounting - or narrow - Credit Controller. Roles can be assigned to the Enterprise or Trading Entities.

Roles are items that can be reused within a model. Each time you create a Role within a Process you are creating a parent Role within the Enterprise or Trading Entity.

Example:

The Customer Services Role is assigned to the Enter Order Stage in the Sales Order Entry Process. This allows application users, who are allocated to the Customer Services Role, to enter sales orders into the system.

Relationship with other elements:

Roles do not explicitly link to or create other items.

3.1.5 Groups

Definition:

Groups are a means of listing items in relevant sections. They are only used to allow greater visibility for the designer and do not have any affect on a run-time system. Each area that uses Groups has one permanent Group called (**unassigned**). This is used as a holder for all data items unless they were created in another Group. You can drag and drop items into and out of the (**unassigned**) Group as required. Unassigned items will still work effectively in the run-time system, since Groups do not affect the run-time system in any way.

Example:

The Sales Process Group contains the Sales Order Entry Process as well as any other sales-focused Processes.

Relationship with other elements:

Groups are used to contain the following items:

- Processes
- Variables
- Tables
- Lookup Lists

3.1.6 Processes

Definition:

The Business Processes (or day-to-day activities) by which a Trading Entity conducts its business whereby nothing is traded. Processes can be internal Processes, utilised to maintain Reference Data, adjust Aggregated data, or to undertake a Process in a defined and controlled fashion, or external Processes where external parties are involved but the item traded is typically dialogue (for example CRM).

A Process is a collection of Process Stages joined by workflow.

Example:

Acme Distribution has a Process called Sales Order Entry. This consists of the following Stages:

- Enter Order
- Print Packing Slip
- Print Order Acknowledgment
- Confirm Despatch
- Print Documents

This Process accurately maps the Business Process of entering orders for Acme Distribution.

Relationship with other elements:

- **Stages** - Stages define the processing steps within a Process, they can be scheduled, have one or more Roles associated with them, and will include one or more Tasks. Refer to page 18 for more information.

3.1.7 Stages

Definition:

A Stage defines a step that comprises one or more Tasks, which can be scheduled within a Process.

Example:

The Print Documents Stage in the Sales Order Entry Process has several Tasks:

- Print Invoice
- Print Labels
- Print Consignment Note

Relationship with other elements:

- **Tasks** - Tasks can only be assigned to Stages. Tasks can be added or changed at the Trading Entity level.
- **Roles** - Roles can only be assigned to Stages. Roles can be added or changed at the Trading Entity level.

3.1.8 Tasks

Definition:

A Task is an activity undertaken within a Stage that is not scheduled but rather initiated when any previous Task within the Stage is completed. Various types of Tasks can be represented including worksheets, reports,

launching external applications (Excel, Word, ERP (Enterprise Resource Planning) program, etc.), manual operations, document operations, updating of Reference Data, sending emails, and so on. Tasks can be reused throughout an Enterprise or Trading Entity. A Task assigned to a Stage is known as a Task instance and the Task it is assigned from is known as the parent Task.

Example:

The Print Document Stage in the Sales Order Entry Process is performed by the system, which means it is performed automatically, without user intervention. In a different system this Task could be assigned to a user who would have to tell the system to print each invoice as it occurs.

Relationship with other elements:

Tasks do not explicitly link to or create other items.

3.1.9 Process Flow

Definition:

The sequence in which Stages are performed within a Process, or Tasks are performed within a Stage.

Process Flow permits consecutive, concurrent, and conditional flow to be defined; once a Process has been initiated the XSOL Process Manager controls the 'what happens next' scenario, using the Roles assigned to the Stages to achieve this.

There are two types of Process Flow:

- **Stage Process Flow** - The order in which Stages are performed in an Process.
- **Task Process Flow** - The order in which Tasks are performed in a Stage.

Each type of Process Flow can be a straight forward A to B to C Process, or a more complicated A to B or C condition-based Process.

Example:

The Sales Order Entry Process, as shown in the following image, contains several Stages and one condition. The condition determines whether an order-acknowledgement is required between the Print Packing Slip and Confirm Despatch Stages.

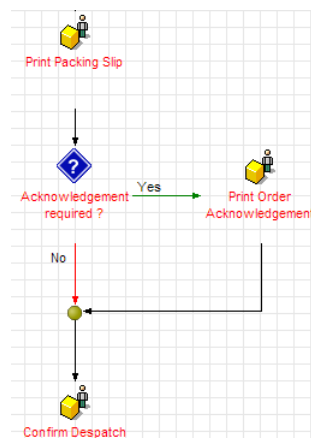


Figure 3.3: Process Flow example

This condition allows sales orders to proceed in two different manners, depending on whether an order-acknowledgement is required.

3.1.10 Tables



Automation Systems Design only

Definition:

There are several classifications of Tables within *XSOL Mapping*: Resources; Resource Extensions; Control Groups; and Relationships. These classifications are utilized to enforce certain principles of ESL and to simplify model construction. A Table is effectively the physical recording of logical groups of Data Fields within a database.

Example:

Acme Distribution has a Customer Resource Table that contains customer information, along with two Resource Extensions, Customer Contacts and Customer Credit Control, that extend on that information.

3.1.11 Resources



Automation Systems Design only

Definition:

The discrete things that an Enterprise employs when undertaking its business, such as Products, Customers, Staff and Bank Accounts. They are typically tangible assets that would appear on the balance sheet.

Example:

The Sales Order Entry Process for Acme Distribution uses two Resources: Customer and Product. These are used to store individual customer and product information.

Relationship with other elements:

- **Datafields** - A Resource is a collection of datafields.

3.1.12 Resource Extensions



Automation Systems Design only

Definition:

A logical group of Data Fields that is implicitly linked to a Resource but held as a separate Table. Each Resource may have many Resource Extensions and each Resource Extension may have a 1-to-1 or 1-to-many relationship to its parent Resource.

Example:

The Customer Resource in the Sales Order Entry Process contains a Customer Contacts Resource Extension. This Table contains extra information about the contacts for a particular customer, including Contact Surname, Forename, and address datafields. This is visible as a separate part of the system, and is able to be searched independently from the Customer Resource.

The Customer Contacts Resource Extension has a 1-to-many relationship with the Customer Resource. This means that the Customer Contacts Table contains it's own UId (Unique Identifier) field(s) that allow it to be searched independently of the Customer Resource.

You can also have a 1-to-1 relationship between a Resource Extension and a Resource. This means that the Resource Extension does not have its own UId field(s) and uses the parent Resource's UId field(s) instead. This type of Resource Extension cannot be searched independently of it's parent Resource. The Customer Credit Control Resource Extension is an example of a 1-to-1 Resource Extension. For example, a 1-to-1 Resource Extension is useful when sensitive information needs to be isolated to another table.

Relationship with other elements:

- **Datafields** - Datafields at the Resource Extension level can only be used by the parent Resource Extension.

3.1.13 Control Groups



Automation Systems Design only

Definition:

A Table that usually comprises a code and description that is used to validate input content on other Tables and then enable grouping of data elements for viewing and/or reporting. These are typically validation- or grouping-type Tables.

Example:

The Acme Distribution system uses a Customer Status Control Group to define that a particular customer can be "Active", "Inactive", or "Stop Credit", and enable a user to search for those types of customers. The Control Group is also used to choose one of these options when a new customer is being entered into the system.

Relationship with other elements:

- **Datafields** - Datafields at the Control Group level can only be used by the parent Control Group.

3.1.14 Relationships



Automation Systems Design only

Definition:

A Table used to hold information that is relevant to one or more Resources and/or Control Groups that can also have optional range parameters specified. A Relationship has three separate segments in its UI: Tables; Ranges; and Datafields. A Relationship must contain a minimum of one Resource or Control Group but can contain more of either table type.

The range parameters are optional and include date, time, datetime (mutually exclusive with the date and time options), days of the week, and the mutually exclusive numeric and percentage breakpoints.

Example:

The Sales Order Entry Process for Acme Distribution contains a Relationship called Customer Pricing. This Table contains links to the Customer and Product Resource Tables and includes a range selection of Effective For and a datafield of Contract Price. This allows the definition of a specific price when a customer purchases the product between the specified dates.

Relationship with other elements:

- **Datafields** - Datafields at the Relationship level can only be used by the parent Relationship.
- **Resources** - Relationships can reference one or more Resources.
- **Control Group** - Relationships can reference one or more Control Groups.

4 How do I use XSOL Mapping?

The following sections illustrate all of *XSOL Mapping*'s functionality. Refer to each section as required.

4.1 Logging on

You need to log on in order to use *XSOL Mapping*.

To log on:

1. Click on the **Start >Programs >XSOL >Mapping** menu option.
2. Click on the radio button next to one of the following options:
 - **Create a new blank model** - Create a new model.
 - **Load an existing model** - Loads an existing model. Click on one of the files that you have previously opened in the available list.



You may not see these options depending on the application settings (refer to page 169 for more information).

3. Click on the **OK** button.

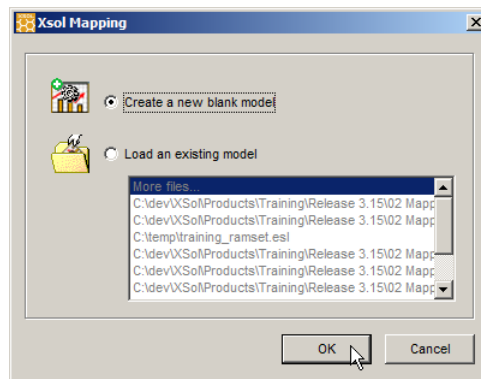


Figure 4.1: Logging on

4.1.1 Logging off

Once you have finished using *XSOL Mapping* you need to log off from the application.

To log off:

1. Click on the **File >Exit...** menu option, or press **Alt+F4**.

4.2 Using XSOL Mapping in a multiple-user environment

Using *XSOL Mapping* in a multiple-user environment is almost the same as using *XSOL Mapping* by yourself. However, there are a few items you need to bear in mind:

- One model (.esl file) can only be opened by one copy of *XSOL Mapping* at a time. If more than one person needs to make changes to the same model, each person needs to open the model at different times.
- You need to set the **Default images folder** option to a network directory that all *XSOL Mapping* users have read/write access to. This directory is used to save all images that are used in User Notes to a central location so that all images are available to all users of a model, irrespective of the machine they are using. Refer to page 169 for more information on how to change the images directory.
- If another person needs to view or alter a model on another machine which does not have access to the **Default images folder** directory, you need to send them both the required model and all images in your images directory. The images need to be saved in that machine's own **Default images folder** directory.

4.3 Using the XSOL health check

The health check function displays your computer's status, both general system information (operating system, computer name, time, logged on username, etc.) and XSOL-specific information (status of installed XSOL applications). You can print this information or send it in an email to XSOL if the XSOL helpdesk requires it.

To use the XSOL health check function:

1. Click on the **Help >About** menu item.
2. Click on the **Health check** button.
3. A wide variety of system information is now displayed.

To email this information to XSOL click on the **Email** button, click on the main body of the email, and press **Ctrl+V** to paste the information into the email. The email can now be sent.

To print this information click on the **Printable version** button and press **Ctrl+P**. Use the **Print** dialog when it appears to print the health check information.

4.4 General functions

There is a number of functions in *XSOL Mapping* that are used throughout the application, such as the Properties Panel and context-sensitive drop-down menus. This section describes those areas.

4.4.1 Windows

Most *XSOL Mapping* windows have the same design and layout. The numbered description refers to the black numbering on the diagram.

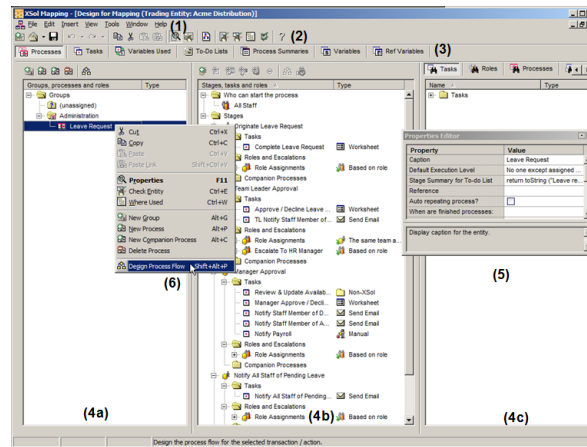



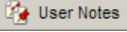
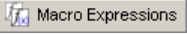
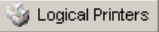

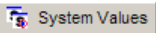
Figure 4.2: XSOL Mapping window description

Description:


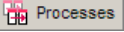


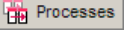


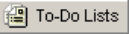
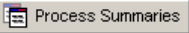
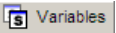





1. **Drop-down menus** - Like many Windows applications, *XSOL Mapping* has a number of drop-down menus available from the top of the window. Only global functions are available from the drop-down menus.
2. **Tool bar** - The Tool bar mimics the global functions available from the drop-down menus. Each button also has a keyboard shortcut, if you prefer the keyboard to the mouse.
3. **Tab bar** - The tab bar is a line of buttons underneath the tool bar. These buttons display different types of information in the same window - refer to page 26 for more information.
4. **Window panes** - Most of *XSOL Mapping*'s windows are divided into three areas:
 - (a) The **left** pane displays a set of information.
 - (b) The **middle** pane displays another set of information.
 - (c) The **right** pane has a variety of reference information that can be useful for the items displayed in the first two panes.
5. **Properties Editor** - The **Properties Editor** is a floating panel that is available in all windows - refer to page 27 for more information.
6. **Options menu** - When you right-click on an item or in a pane you open the **Options** menu. The **Options** menu provides another means of accessing all the functions that are available for the selected item or pane.
7. **Status bar** - The status bar displays messages about what is happening in *XSOL Mapping*. You can use the status bar to display descriptions of each function in the application.


4.4.2 Navigation

XSOL Mapping uses a series of designer buttons to navigate to required functions. You can choose which designer buttons you want to display, and in which order (refer to page 27 for more information). The **Design Enterprise** window, which is the first window that opens when you load a model, contains by default the following designer buttons:

-  Roles
-  User Notes
-  Macro Expressions
-  Logical Printers
-  Document Locations
-  System Values

The **Design Enterprise** window contains three buttons that access the major functionality areas of *XSOL Mapping*:

-  - Opens the **Design for Mapping** window, which has by default the following designer buttons:
 -  Processes
 -  Tasks
-  - Opens the **Design for Automation** window, which has by default the following designer buttons:
 -  Processes
 -  Tasks
 -  Variables Used
 -  To-Do Lists
 -  Process Summaries
 -  Variables
 -  Ref Variables
 -  Tables
 -  Control group entries
 -  Maintenance Tasks
 -  Lookup Lists

For ease of use, you can often access the same functionality from different windows. For example, you can access the  **Tasks** designer button from both the **Design for Mapping** and the **Design for Automation** windows. You can also choose which designer buttons you want to display, and in which order (refer to page 27 for more information).

To navigate in XSOL Mapping:

1. Click on one of the three buttons, detailed above, displayed in the **Design Enterprise** window.
2. Click on one of the designer buttons within the opened window as detailed above.



Press **Ctrl+D** to quickly return to the **Design Enterprise** window.

4.4.3 Customising designer buttons

For each of the main *XSOL Mapping* windows (**Design for Mapping** and **Design for Automation**) you can choose to display, hide, or re-order any of the available designer buttons.

To customise the designer buttons:

1. Click on the **Tools >Customise >Show** menu item.
2. Click on the window that you want to change in the left pane.
3. To display or hide a designer, select or remove the selection from the tick box next to the required designer.

To order the designer list, click on the required designer and click on the  or  button.

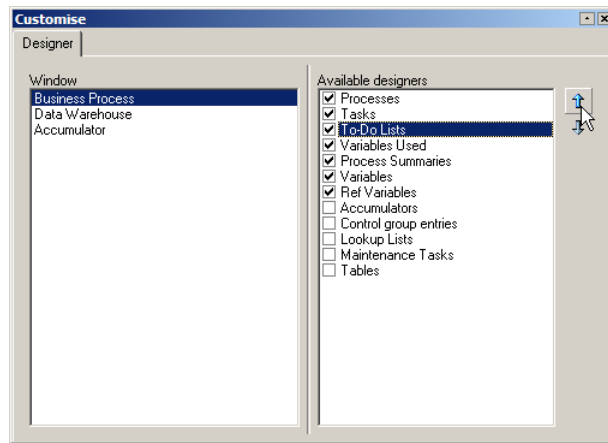


Figure 4.3: Ordering the designer list

4.4.4 Properties Editor

The **Properties Editor** is a floating panel that is available in all windows. This panel allows you to view and edit the properties of any particular item that you have selected in the active window. The **Properties Editor**, when open, is always displayed on top of any other open windows. Because this can be intrusive, there are a number of keyboard shortcuts that allow you to quickly open, close, and "roll up" the **Properties Editor**.

- **F11** - Open or close the **Properties Editor**.
- **Shift+F11** - "Roll" the window up so that only the title bar is visible. Press Shift+F11 again to display the full panel again.

The **Properties Editor** can contain a wide variety of information. However, it will always contain the following items:

- **Caption** - The name of the selected item.
- **Runtime Hint** - Displays information to the user in an *XSOL Automation* system. All runtime hints that are added to visible elements (such as widgets on a Worksheet) are displayed in an *XSOL Automation* system.
- **User Notes** - Allows you to enter various types of additional information about the selected item. Refer to page 145 for more information.

- **Version information** - Displays when the item, or one of the items it contains, was last modified (**Last modified by**) and by whom (**Last modified at**, based on the Window's user name logged in at the time). The **Version note** property is a permanent User Note - refer to page 145 for more information.
- **Designer comments** - Allows you to enter designer-specific information into a simple text editor. Designer comments are not included in the generated documentation.
- **Help URL** - Not currently used.

You can set the visual appearance used for the text in the Properties Editor (refer to page 172 for more information).

4.4.5 Drop-down menus

XSOL Mapping uses drop-down menus in all windows. There are two types of menus:

- **Context-sensitive** - The options available within the menu change depending on what window you are using.
- **Static** - The options available within the menu are always the same.

The available menus are detailed below:

- **File** - A static menu that enables you to manipulate models.
- **Insert** - A context-sensitive menu that enables you to insert items into a model.
- **Process** - A context-sensitive menu that enables you to perform various actions. Not available in all windows.
- **View** - A context-sensitive menu that enables you to open windows or change the viewing area of an existing window.
- **Tools** - A static menu that enables you to validate and generate output from a model.
- **Window** - A static menu that enables you to manipulate and move between open windows.
- **Help** - A static menu that gives you access to comprehensive help information on using *XSOL Mapping*.

To use a menu:

1. Click on the required menu from the top of the *XSOL Mapping* window.
2. Click on the option that you want to use in that drop-down menu.

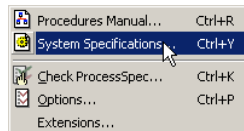


Figure 4.4: Using a menu

4.4.6 Resizing items

You can resize many items in *XSOL Mapping*, including window panes, columns, Process Flow nodes, and Widgets on a Worksheet. Resizing an item only changes its visual appearance and does not influence the rest of the model.



Refer to page 34 for more information on how to set the maximum and default node sizes for Process Flow nodes.

To resize an item:

1. To resize a pane or column:
 - (a) Move the mouse over the border pane or column until the cursor becomes a vertical line with two arrows facing opposite directions.
 - (b) Click and drag the border to the required position, then let go of the mouse button.

To resize a Process Flow node or Widget:

- (a) Press and hold down the **Shift** key.
- (b) Click on the gray outline on the side of the node that you want to resize and drag the boundary box to the required size.

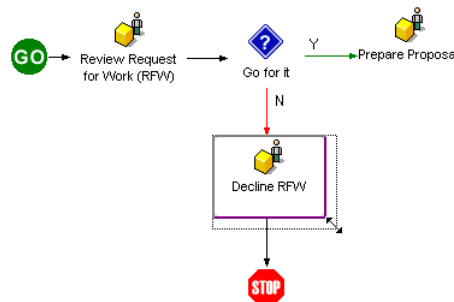


Figure 4.5: Resizing a Process Flow node

4.4.7 Sorting columns

You can sort most columns in *XSOL Mapping* alphabetically. For example, you can sort Tasks either by their names or types. This allows you to find specific items quickly and efficiently.

To sort columns:

1. Find the column that you want to sort.
2. Click on the column header that you want to sort. The following icons indicate how the columns have been sorted:
 - - The column is sorted in ascending sequence - alphabetical order, 1-10.
 - - The column is sorted in descending sequence - reverse-alphabetical order, 10-1.

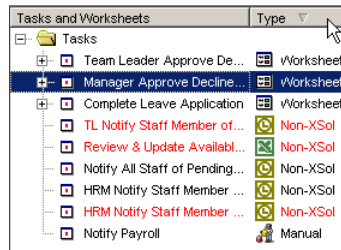






Figure 4.6: Sorting a columns

4.4.8 Undoing and redoing changes

XSOL Mapping enables you to undo and redo any changes that you make. You can undo and redo any changes that are made in a model, with the following exceptions:

- You are limited to the number of undo items that are stored in memory by an option (refer to page 169) for more information.
- You cannot undo the changes made by the **Remove Versions** menu option (refer to page 38 for more information).

To undo and redo changes:

1. Click on the  button to undo the last change. If you want to undo a specific change which is not the most recent change, click on the small down arrow next to the  button and click on the required change.
2. Click on the  button to re-apply the last change. If you want to re-apply a specific change (and all changes up to it in the list) which is not the most recent removed change, click on the small down arrow next to the  button and click on the required change.

4.4.9 Copying and pasting

You can copy and paste items in a model (one copy of *XSOL Mapping* open) or between models (more than one copy of *XSOL Mapping* open), as required. The following rules apply:

- You can copy any item in a particular level of the model and paste within that level. All items that are linked to that item are copied as well. For example, you can copy and paste a Resource, including all linked datafields, within the **Design for Automation** window, or a Condition within the **Design Process Flow** window.
- You can copy any item in a particular level of the model and paste within that level in another Trading Entity. *XSOL Mapping* will ask you to choose whether to copy any linked items from the source Trading Entity or to use existing items in the destination Trading Entity, if any.

There are two different types of paste function:

- **Paste** - Inserts the copied item. If the copied item is an instance of a global item (such as a Task, Role, or Variable), the global item is copied as well. For example, using paste on a Task node in Process Flow will create a copy of the parent Task as well as the instance of that Task in the Process Flow.

- **Paste Link** - Inserts the copied item. If the copied item is an instance of a global item (such as a Task or Role), the global item is not copied. For example, using paste on a Task node in Process Flow will only create a copy of the instance of that Task in the Process Flow.

The last item that has been copied to the clipboard is highlighted in blue, which provides a visual reference of what item is in the clipboard.

To copy and past within the same Trading Entity or Enterprise:

1. Click on the item that you want to copy and press **Ctrl+C**. This places the item in the clipboard, and highlights the item in pale blue.
2. Press **Ctrl+V** to paste the item into the same location. You can paste the item multiple times, if required.

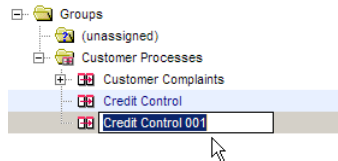


Figure 4.7: Copying and pasting a Resource

To copy and paste between Trading Entities or a Trading Entity and Enterprise:

1. Click on the item that you want to copy and press **Ctrl+C**. This places the item in the clipboard, and highlights the item in pale blue.
2. Open the required location in the destination Trading Entity.
3. Press **Ctrl+V** to paste the item. You can paste the item multiple times, if required.
4. The **Resolve Dependencies** window will open if there are any dependencies in the copied item. Click on the **Map to** column next to the required item and click on the required dependency to use.



The **Copy from source** option creates a new item based on the original item from the source Trading Entity and remembers the fact that it came from that item. All other options are items in the destination Trading Entity of the same type, and creates a new item based on this existing item.

5. Repeat for all dependencies, as required. Click on the **OK** button when you have finished.

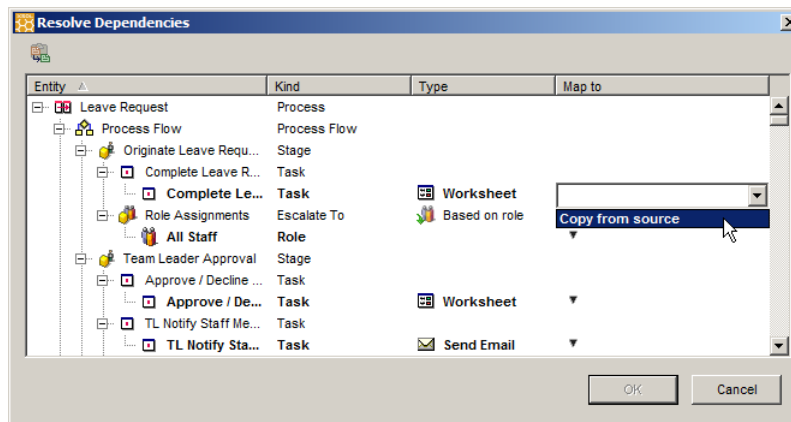



Figure 4.8: Resolving dependencies with a pasted Process

6. If you have a lot of dependencies and you want to resolve them quickly, click on the  button.
7. The default settings for the **Auto Resolve Dependencies** window are **Narrow**, **Copy all entities that cannot be automatically mapped**, and **Map their dependencies**. This is the recommended setting for the majority of situations. Only change these settings if you are sure of what you are doing - please read the following descriptions, if required.
 - **Narrow** - Only matches items that have the same source ID.
 - **Wide** - Only match entities that are of the same type and subtype (for example, a "Send Email" Task Type) and have the same caption name.
 - **Custom** - Only match entities based on one or more criteria. **Source GUID** is the same as **Narrow**, while **Caption** is the same as **Wide**. **External ID** matches entities that have the same **External ID** property (refer to page 27 for more information).
 - **Copy all entities that cannot be automatically mapped** - For all items that cannot be mapped using the previous search choices, create a new item based on the original item from the source Trading Entity and remember the fact that it came from that item. This is the same as the **Copy from source** option.
 - **Map their dependencies** - If the item has any dependencies, copy the dependencies as well.
8. Click on the **Map** button.

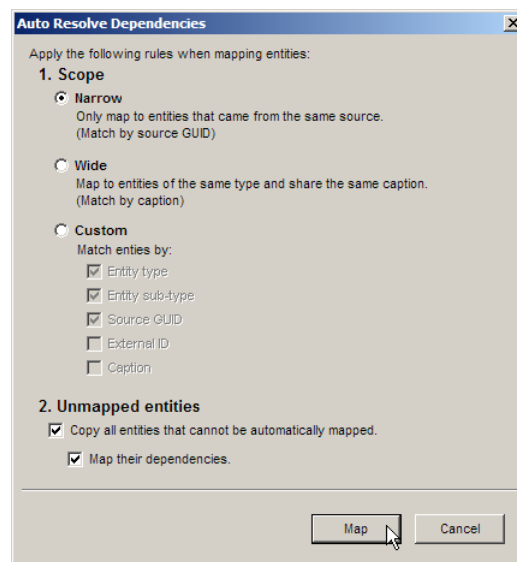


Figure 4.9: Automatically resolving dependencies when copying and pasting

9. Click on the **OK** button.





4.4.10 Groups

Groups are a means of listing items in relevant sections. They are only used to allow greater visibility for the designer and do not have any affect on a run-time system. Each area that uses Groups has one permanent Group called (**unassigned**). This is used as a holder for all data items unless they were created in another Group. You can drag and drop items into and out of the (**unassigned**) Group as required. Unassigned items will still work effectively in the run-time system, since Groups do not affect the run-time system in any way.

You can create Groups for the following areas:

- Processes
- Variables
- Tables
- Lookup Lists

To add a Group:

1. Open the required window.
2. Click on one of the following buttons (depending on what window you are in):
 -  - Add Process Group button.
 -  - Add Table Group button.
 -  - Add Variable Group button.
 -  - Add Lookup List Group button.
3. Enter the name of the Group and press the **Enter** key to create another item or press the **Tab** key to only create this item.
4. Add or change any properties in the **Properties Editor**, if required.

Changing a Group

Changing a Group involves adding or changing any properties for this item.

To change a Group:

1. Click on the Group in the required window.
2. Change any properties in the **Properties Editor**, if required.

4.4.11 Deleting items

You can delete an item in *XSOL Mapping* at any time.

To delete any item:

1. Navigate to the window that contains the item that you want to delete.
2. Click on the required item.
3. Press the **Delete** key or click on the nearest button with a red circle on it.
4. If the item is used elsewhere in the model, a dialog window will appear informing you of this fact and where the item is used. You need to remove the references to the item elsewhere in the model before you can delete the item.
5. Click on the **Yes** button in the **Confirm** window.

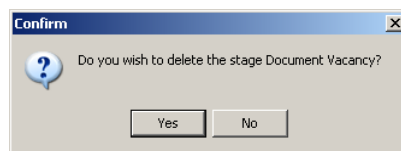


Figure 4.10: Deleting a Stage

4.5 Designing a model

A model is an XML (Extensible Markup Language) document based on ESL (Enterprise System Logic) terminology that defines the business model for a specific Enterprise in terms of its Trading Entities, Processes, Stages, Tasks, Reference Data, Controls and Data Fields. *XSOL Mapping* automatically loads the last edited model when the application is opened, if that option has been selected. Only one model can be edited by *XSOL Mapping* at one time.

Model properties:

All properties apply globally to the model. However, you can always change individual properties to a different value, if required. Pre-existing values are not changed if a global value is changed.

- **Default Unit of Measure** - Specifies the standard unit of measurement for all properties in the model.
- **Paper Size** - Specifies the default paper size used for all Process Flow diagrams.
- **Default rotation for flow diagrams** - Specifies the orientation used for all Process Flow diagrams. This property is used when generating Word documentation - refer to page 151 for more information.
- **Show stage icon** - Refer to page 74 for more information.
- **Show stage border** - Refer to page 74 for more information.
- **Show task icon** - Refer to page 74 for more information.
- **Show task border** - Refer to page 74 for more information.
- **Reference Caption Position** - A reference caption is a user-defined labeling system that is displayed on Stages and Tasks in Process Flow diagrams. This property specifies where the reference caption is placed on the Process Flow diagrams - refer to page 100 for more information. If **None** is selected reference captions are turned off. You cannot change the reference caption position for individual Stages or Tasks.
- **Reference Caption Separator** - Specifies the separator to put between Process, Stage, and Task reference captions on Task Process Flow diagrams. Only available if the **Reference Caption Position** option has been activated.
- **Author** - Enter the name of the primary author of this model. Optional.
- **Maximum flow node width** - Enter the maximum allowed width for any Process Flow diagram node.
- **Maximum flow node height** - Enter the maximum allowed height for any Process Flow diagram node.
- **Default width for annotated flow nodes** - Enter the default width for any Process Flow diagram node that displays a caption on the diagram. This only applies to new nodes, not existing ones.
- **Default height for annotated flow nodes** - Enter the default height for any Process Flow diagram node that displays a caption on the diagram. This only applies to new nodes, not existing ones.
- **Default width for other flow nodes** - Enter the default width for any Process Flow diagram node that does not display a caption on the diagram. This only applies to new nodes, not existing ones.
- **Default height for other flow nodes** - Enter the default height for any Process Flow diagram node that does not display a caption on the diagram. This only applies to new nodes, not existing ones.
- **Save or Reset application defaults** - Allows you to set the current default icons as the permanent default icons - refer to page 92 for more information.

- **Reset individual node icons** - Allows you to re-set all currently existing icons to the application defaults - refer to page 92 for more information.
- **Default caption for new 'True' links** - Enter the default caption for all true links from Condition nodes - refer to page 85 for more information. This will change existing link captions (unlike other default settings that only apply to new items, not existing ones).
- **Default caption for new 'False' links** - Enter the default caption for all false links from Condition nodes - refer to page 85 for more information. This will change existing link captions (unlike other default settings that only apply to new items, not existing ones).
- **Draw 'True' links first?** - If ticked, true links are created first from Condition nodes - refer to page 85 for more information.
- **Default stage flow 'Print in documentation'** - If ticked, Task Process Flow is automatically included in generated documentation. This only applies to Stages when they are created - it will not change existing Stages.
- **Default 'show role assignments' value** - If ticked, Role information is displayed on Process Flow diagrams - refer to page 96 for more information.
- **Default 'true' value** - Specifies the default true value used by any Binary User Note or Variable - refer to page 145 and/or page 128 for more information.
- **Default 'false' value** - Specifies the default false value used by any Binary User Note or Variable - refer to page 145 and/or page 128 for more information.
- **Default 'null' value** - Specifies the default null value used by any Binary User Note or Variable - refer to page 145 and/or page 128 for more information.
- **Default flow control icons** - Specifies the default icon used by any Process Flow node - refer to page 92 for more information.
- **Default task icons** - Specifies the default icon used by any Process Flow Task node - refer to page 92 for more information.

To create a model:

1. Open *XSOL Mapping*. Refer to page 22 if you do not know how to open the application.
2. Click on the **File >New Model** menu option, or press **Ctrl+N**. This automatically opens the **Design Enterprise** window and creates a new model with a blank Enterprise.

You will now need to start adding to the model (refer to page 39 and page 48 for more information).

4.5.1 Saving a model

Make sure that you save your model often to ensure that you do not accidentally lose any changes that you have made. You can also save the current model to another file as a backup copy, if required.

To save a model:

1. Click on the **File >Save** menu option, or press **Ctrl+S** to save the current model XML (Extensible Markup Language) file. If you need to save the model as a new XML file, go to the next step.
2. Click on the **File >Save As...** menu option, or press **Shift+Ctrl+S** to save the current model as a new XML file.

3. Choose where you want to save the file, enter the name of the file in the **File name** field, and click on the **Save** button.

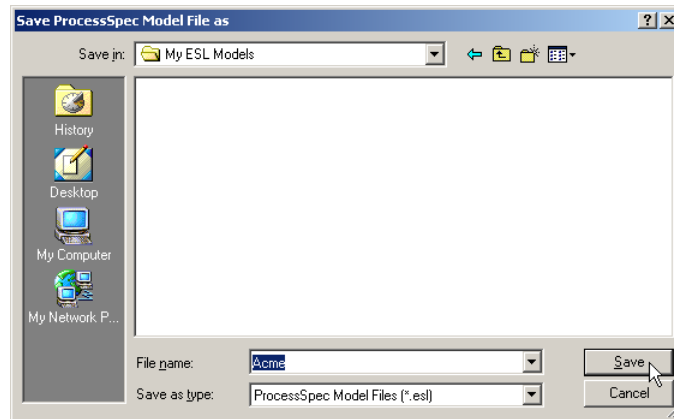


Figure 4.11: Saving a model

4.5.2 Packaging a model

You need to package a model before you can send it to another *XSOL Mapping* user. Packaging compresses the model and associated User Notes images and custom Non-XSOL Task icons into one file, which can then be emailed or otherwise sent to someone else.

To package a model:

1. Click on the **File > Save As...** menu option, or press **Shift+Ctrl+S**.
2. Click on the **ESL Model Packages (*.epk)** option in the **Save as type** field.
3. Choose where you want to save the file, enter the name of the file in the **File name** field, and click on the **Save** button.

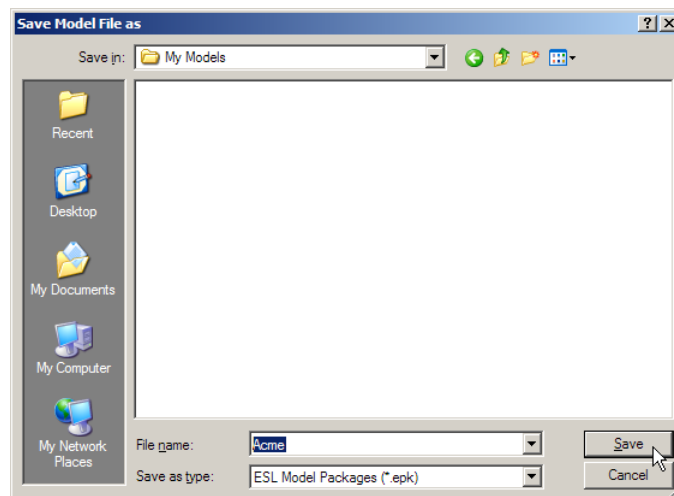


Figure 4.12: Packaging a model

4. You can now email the model to another *XSOL Mapping* user.

Unpacking a model

Packed models end in the extension .epk. These packed models include all User Notes images and custom Non-XSOL Task icons into one file. When unpacked, all images are copied to the **Default images folder** and the model is saved as an .esl file.

To unpack a model:

1. Click on the **File >Load...** menu option, or press **Ctrl+O**.
2. Find the location of the .epk file that you want to load and double-click on it.
3. Choose where you want to save the unpacked .esl file, enter the name of the file in the **File name** field, and click on the **Save** button.
4. The **Design Enterprise** window is opened automatically. You can now edit the model.

4.5.3 Loading a model

You need to load a model in order to work on it in *XSOL Mapping*. Loading a model while working on another model will prompt *XSOL Mapping* to ask you if you want to save the current model, and then will close the current model before opening the new model.

To load a model:

1. Click on the **File >Load...** menu option, or press **Ctrl+O**.
2. Find the location of the model that you want to load and double-click on it.

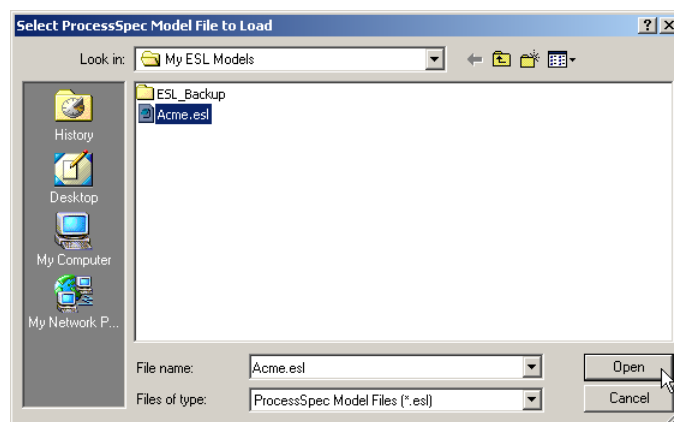


Figure 4.13: Loading a model

3. If you have not saved the currently open model, you are now asked if you want to save it. Click on one of the available buttons, as required.
4. The **Design Enterprise** window is opened automatically. You can now edit the model.

4.5.4 Restoring a model

XSOL Mapping provides you with an automatic method of backing up your model. This is the **Always backup the model before saving** tick box in the **Options** window (refer to page 169 for more information about setting this option). When set, this creates an _Backup\ directory in the same folder as the saved

model. Every time you save the model (the **File >Save** or **File >Save As...** menu options), a new .ebk file is created with the name of the model and the save number, for example **Acme.0025.ebk**.

You can restore any of the backup files simply by copying the file to another folder and renaming the file.

To restore a model:

1. Open the _Backup\ directory in the same folder as the model that you need to restore.
2. Choose the file with the same name as your model that you would like to restore. The .ebk file with the highest number represents the last time the model was saved, and is generally the file that you want.

Name	Size	Type
Acme.0001.ebk	229 KB	EBK File
Acme.0002.ebk	5 KB	EBK File
Acme.0003.ebk	5 KB	EBK File

Figure 4.14: Choosing the correct backup file

3. Copy the file to the required folder.
4. Change the name of the file to the same name as the original model.



You need to change the extension from .ebk to .esl.

5. You can now load the model.

4.5.5 Upgrading a model

When you upgrade to a new version of *XSOL Mapping*, you may find that your models do not run under the new version. Fortunately, the new version of *XSOL Mapping* will automatically detect if a model needs to be upgraded and ask you if you want to upgrade. Click on the **Yes** button to upgrade the model automatically.

4.5.6 Removing model version information



Automation Systems Design only

Usually, when a model is committed to an *XSOL Automation* system, an updated version of the model is returned by the XPM (XSOL Process Manager) which contains version information. Version information indicates which version of the running system this model is, and contains a complete history of all previous versions. Version information is cumulative - it adds to existing version information each time a model is committed.

In the testing process, it can be useful to remove all version information from a model when you are ready to create a live system, effectively starting at version one.



Do not use this function if you are not sure of what you are doing. You cannot undo any changes made by the **Remove Versions** menu option.

To remove model version information:


1. Click on the **Tools >Remove Versions** menu option.
2. Click on the **Yes** button in the confirmation window.

4.6 Designing an Enterprise

The Enterprise is the top node of the hierarchy. An Enterprise is a container for the Business Processes of one or more Trading Entities. The Enterprise has the same capabilities as a Trading Entity, thus Roles, Reference Data, and Processes can all be defined.

For an expanded definition please refer to page 16 for more information.

To create a new Enterprise:

1. Click on the **File > New Model** menu option, or press **Ctrl+N**. This automatically opens the **Design Enterprise** window and creates a new model with a blank Enterprise.
2. Click on  Enterprise in the Hierarchy list.
3. Change the name of the Enterprise caption in the **Properties Editor** to the name of the Enterprise.
4. Add or change any properties in the **Properties Editor**, if required.

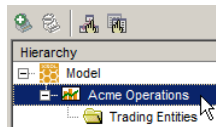


Figure 4.15: Designing an Enterprise

4.6.1 Changing an Enterprise

Changing an Enterprise can involve altering the Enterprise's name, which will change the title page of the generated documentation and any User Notes that you have added to the Enterprise's properties.

To change an Enterprise:

1. Click on the Enterprise at the top of the Hierarchy list.
2. Change any properties in the **Properties Editor**, if required.

4.6.2 Deleting an Enterprise

You cannot delete an Enterprise once one has been created, as the Enterprise is the starting point for the entire model. However, you can change the name and User Notes attached to the Enterprise (refer to page 39 for more information).

4.6.3 Designing a Trading Entity

A discrete entity within the Enterprise through which business is conducted; each Trading Entity will have its own Roles, Reference Data, Processes, and Arithmetic data aggregation defined.

For an expanded definition please refer to page 16 for more information.


Trading Entity properties:

All Trading Entity properties are used when archiving, removing, and deleting Processes, and in Process Summaries - refer to page 48 and page 74 for more information.

- **Start month of financial year** - Select the month that marks the beginning of the financial year for this Trading Entity from the drop-down list.

- **Time Zone** - Select the required time zone for this Trading Entity from the drop-down list.
- **First Day of the Week** - Select the day that marks the beginning of the working week for this Trading Entity from the drop-down list.

To add a Trading Entity:

1. Click on the  button at the top of the **Design Enterprise** window.
2. Enter the name of the Trading Entity and press the **Enter** key to create another item or press the **Tab** key to only create this item.
3. Add or change any properties in the **Properties Editor**, if required.

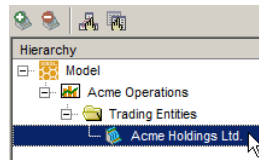


Figure 4.16: Adding a Trading Entity

Changing a Trading Entity

Changing a Trading Entity involves adding or changing any properties for this item.

To change a Trading Entity:

1. Click on the Trading Entity in the **Design Enterprise** window.
2. Change any properties in the **Properties Editor**, if required.



4.6.4 Using Logical Printers



Automation Systems Design only

Logical Printers are the printers that are available from an *XSOL Automation* system for printing items as a part of running Processes.

To add a Logical Printer:

1. Click on the  **Logical Printers** designer button in the **Design Enterprise** window.
2. Click on the Enterprise or Trading Entity to which you want to add a Logical Printer.
3. Click on the  button.
4. Enter the name of the Logical Printer and press the **Enter** key to create another item or press the **Tab** key to only create this item.



Do not use any of Python's reserved keywords as names, as they might cause an error in Python - refer to page 198 for more information.

5. Choose one of the following options in the **Scope** field in the **Properties Editor**:
 - **Children** - The Logical Printer, if created at the Enterprise level, can be used by any Trading Entities within that Enterprise.

- **Local** - The Logical Printer can only be used in the Enterprise or Trading Entity in which it was created.
6. Add or change any properties in the **Properties Editor**, if required.



Figure 4.17: Adding a Logical Printer

Changing a Logical Printer



Automation Systems Design only

Changing a Logical Printer involves adding or changing any properties for this item.

To change a Logical Printer:

1. Click on the required Logical Printer in the **Design Enterprise** window.
2. Alter the caption, **Scope** field, or any User Notes in the **Properties Editor**, if required.



Figure 4.18: Changing a Logical Printer



4.6.5 Using Document Locations



Automation Systems Design only

Document Locations are the document storage areas that are available from an *XSOL Automation* system for manipulating documents as a part of running Processes.

To add a Document Location:

1. Click on the  **Document Locations** designer button in the **Design Enterprise** window.
2. Click on the Enterprise or Trading Entity to which you want to add a Document Location.
3. Click on the  button.
4. Enter the name of the Document Location and press the **Enter** key to create another item or press the **Tab** key to only create this item.



Do not use any of Python's reserved keywords as names, as they might cause an error in Python - refer to page 198 for more information.

5. Choose one of the following options in the **Scope** field in the **Properties Editor**:
 - **Children** - The Document Location, if created at the Enterprise level, can be used by any Trading Entities within that Enterprise.
 - **Local** - The Document Location can only be used in the Enterprise or Trading Entity in which it was created.

6. Add or change any properties in the **Properties Editor**, if required.



Figure 4.19: Adding a Document Location

Adding a limit to a Document Location






Automation Systems Design only

You can add a limit to a Document Location using a Document Variable. You can select a specific Document Location and limit it to a specific subfolder (either using a Variable or plain text). Limits restrict what areas an *XSOL Automation* user can access when using a run-time system.

Limit properties:

- **Limit to Document Location** - Used to select the required Document Location for the limit.
- **And Limit to Folder within Location** - Used to determine how the Document Location is limited and to what extent. Choose one of the following options:
 - Any Folder or File - No limitation is applied.
 - From variable - Determine the required limitation from the Variable selected in the **Get Folder From** field.
 - Specify value - Enter the path to the limited folder in the **Input Folder Name** field. You need to start the path with a slash (/), such as **/Samples**.

To add a limit to a Document Location:

1. Click on the  **Variables** designer button in the **Design for Automation** window.
2. Click on the required Document Variable.
3. Click on the  button to create a new Business Rule, if one does not already exist. Re-name the Business Rule to something appropriate.
4. Click on the  button to create a new limit.
5. Add or change any properties in the **Properties Editor**, if required.

Property	Value
Limit to Document Location	A Document Location
And Limit to Folder within Locati...	Specify value
Input Folder Name	/Samples

Figure 4.20: Adding a limit to a Document Location

Changing a Document Location



Automation Systems Design only

Changing a Document Location involves adding or changing any properties for this item.

To change a Document Location:

1. Click on the required Document Location in the **Design Enterprise** window.
2. Change any properties in the **Properties Editor**, if required.



Figure 4.21: Changing a Document Location

4.6.6 Using System Values



Automation Systems Design only

System Values are Variables that are automatically generated in *XSOL Automation* and can be used in Process Summaries or Expressions (refer to page 74 and page 138 for more information about how to use System Values.). There are three types of System Values:

- **Global** - System Values related to the model, Enterprise, and Trading Entities.
- **Process Related** - System Values related to Processes, Stages, and Tasks.
- **Reference Data Related** - System Values related to Tables and datafields.

Global

- **Current user** - Returns the user name of the logged on user.
- **Enterprise name** - Returns the enterprise name of the model.
- **Model committed at** - Returns the date and time the model was committed.
- **Model version** - Returns the model version number.
- **Trading entity name** - Returns the name of the Trading Entity.
- **Trading entity version** - Returns the version number of the Trading Entity.

Process Related

- **Process archived?** - Returns **yes** if the Process is archived, and **no** if it is not.
- **Process finished at** - Returns the date and time the Process was finished.
- **Process initiated by Role** - Returns the Role name that initiated the Process.
- **Process initiated by Team** - Returns the Team name that initiated the Process.
- **Process last updated at** - Returns the last time the Process was updated.
- **Process last updated by user** - Returns the last user to update the Process.
- **Process initiated by user** - Returns the user name that initiated the Process.
- **Process instance id** - Returns the GUID (Globally Unique Identifier) of the Process instance.
- **Process manually initiated?** - Returns **yes** if the Process was manually initiated, and **no** if it was not.

- **Process name** - Returns the name of the Process.
- **Process started at** - Returns the date and time the Process was started.
- **Process state** - Returns one of the following states:
 - Initiated
 - In Progress
 - Terminated
 - Completed
- **Process last updated at** - Returns the date and time the Process was last changed.
- **Process last updated by user** - Returns the user name that last changed the Process.
- **Process version** - Returns the version number of the Process.
- **Stage completion due** - Returns the date and time by which a Stage needs to be completed.
- **Stage escalation level** - Returns the Stage's current escalation level.
- **Stage finished at** - Returns the date and time the Stage was finished.
- **Stage name** - Returns the name of the Stage.
- **Stage started at** - Returns the date and time the Stage was started.
- **Stage state** - Returns one of the following states:
 - Initiated
 - In Progress
 - Terminated
 - Completed
- **Stage undertaken by role** - Returns the Role name that initiated the Stage.
- **Stage undertaken by team** - Returns the Team name that initiated the Stage.
- **Stage undertaken by user** - Returns the user name that initiated the Stage.
- **Stage version** - Returns the version number of the Stage.
- **Task finished at** - Returns the date and time the Task was finished.
- **Task name** - Returns the name of the Task.
- **Task started at** - Returns the date and time the Task was started.
- **Task state** - Returns one of the following states:
 - Initiated
 - In Progress
 - Terminated
 - Completed
- **Task Type** - Returns the current Task type.
- **Task undertaken by role** - Returns the Role name that initiated the Task.

- **Task undertaken by team** - Returns the Team name that initiated the Task.
- **Task undertaken by user** - Returns the user name that initiated the Task.
- **Task version** - Returns the Task version number.

Reference Data Related

- **Created at** - Returns the date and time the record was created.
- **Deleted at** - Returns the date and time the record was deleted.
- **Last updated at** - Returns the date and time the record was last changed.
- **Record created by user** - Returns the user who created the record.
- **Record deleted by user** - Returns the user who deleted the record.
- **Record instance id** - Returns the GUID (Globally Unique Identifier) of the record instance.
- **Record instance version** - Returns the version number of the record. The record number is incremented by one every time the record is changed.
- **Record last updated by user** - Returns the last user to update the record.
- **Table definition version** - Returns the version number of the table. The record number is increased every time the table definition is changed.
- **Table name** - Returns the name of the Table.

4.6.7 Defining User Notes



You can define an unlimited number of User Note fields with their own captions, and you can specify which of them will be included in the generated documentation. Each User Note has a defined type, such as Numeric or Word, which allows you to enter specific types of information. Any item in *XSOL Mapping* can have one or more User Notes added to it, and you can specify whether a User Note is specific to a particular item or generally available for all items.

The following User Note types are available:

- **Binary** - Used when only two value are allowed (also known as Boolean). You can set the following type-specific fields:
 - True Value - The true value of the User Note.
 - False Value - The false value of the User Note.
 - Null or unknown value - The text to display if the User Note has no value.
- **Date** - Used to display dates. You can set date formats using the Input Mask and Display Mask fields.
- **Datetime** - Used to display dates and times. You can set date and time formats using the Input Mask and Display Mask fields.
- **Word** - Used for to enter information into your User Notes with all the functionality of a word processor. You can change font, colors, highlight text, insert bullet points, and insert images and links to external files (refer to page 147 for more information about how style information is stored).
- **Memo** - Used for entering general alphanumeric characters. Memo User Notes are not limited in length and can span multiple lines of text (unlike Notational User Notes).

- **Notational** - Used for entering general alphanumeric characters. Notational User Notes are limited to 120 characters in length and one line of text - multiple lines are not allowed. You can set notational formats using the Input Mask field. You can set the following type-specific field:
 - Length - The length, in characters, of the field. Mandatory.
- **Numeric** - Used for entering numeric characters. You can set numeric formats using the Input Mask and Display Mask fields. You can set the following type-specific fields:
 - No of Integer Digits - How many digits are allowed before the decimal place. Mandatory.
 - No of Decimal Digits - How many digits are allowed after the decimal place. Mandatory.
 - Allow Zero - Indicate whether zero is an acceptable value.
 - Allow Positive - Indicate whether positive numbers are acceptable values.
 - Allow Negative - Indicate whether negative numbers are acceptable values.
 - Ranges - Specify what range of numbers are acceptable values. Use two periods (..) to indicate a range between two numbers, and commas (,) to separate ranges. For example, **-1..10,12..50** will allow all numbers between -1 and 10, and between 12 and 50.
- **Percent** - Used for entering percentile information. You can set percentile formats using the Input Mask and Display Mask fields. You can set the following type-specific fields:
 - No of Decimal Digits - How many digits are allowed after the decimal place. Mandatory.
 - Allow Zero - Indicate whether zero is an acceptable value.
 - Allow Positive - Indicate whether positive numbers are acceptable values.
 - Allow Negative - Indicate whether negative numbers are acceptable values.
 - Ranges - Specify what range of numbers are acceptable values. Use two periods (..) to indicate a range between two numbers, and commas (,) to separate ranges. For example, **-1..10,12..50** will allow all numbers between -1 and 10, and between 12 and 50.
- **Scientific** - Used for numeric values that may vary in size. For example, use a Scientific datafield when users need to enter numeric information which has an unknown number of decimal places. You can set scientific formats using the Input Mask and Display Mask fields.
- **Time** - Used to display time values. You can set time formats using the Input Mask and Display Mask fields.

To define a User Note:

1. Click on the  **User Notes** designer button in the **Design Enterprise** window.
2. Click on the  button.
3. Enter the name of the User Note and press the **Enter** key to create another item or press the **Tab** key to only create this item.
4. Select the type of User Note from the **User note type** drop-down list property. Enter any specific information for this User Note in the **Properties Editor**.
5. Select the tick box in the **Print in Documentation** column if you want the User Note to be included in generated documentation.
6. By default all item categories in the middle pane are selected, which means that the User Notes is available for all items in the model. If the User Note is not to be used by all items, deselect the items in the middle pane to which the User Notes does not belong.

7. Repeat the previous steps for as many User Notes as you require. All User Notes are optional, and are only defined or used when they are needed.

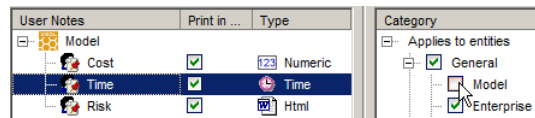



Figure 4.22: Defining a User Note

4.7 Designing Business Processes

Business Processes are the core element of the model. This area maps out what Business Processes are used by the business, what actions are required for a particular outcome, when an action needs to occur, and who performs that action.

To design Business Processes:

1. Click on the  button in the **Design Enterprise** window.
2. You are now in the **Design for Mapping** window. Refer to any of the following sections for more information about designing Business Processes.

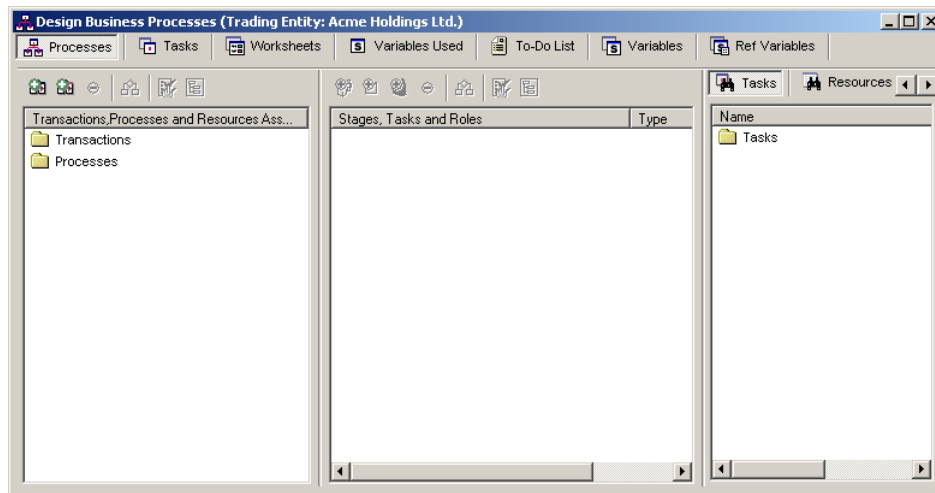


Figure 4.23: Designing Business Processes

4.7.1 Using Processes

The Business Processes (or day-to-day activities) by which a Trading Entity conducts its business whereby nothing is traded. Processes can be internal Processes, utilised to maintain Reference Data, adjust Aggregated data, or to undertake a Process in a defined and controlled fashion, or external Processes where external parties are involved but the item traded is typically dialogue (for example CRM).

A Process is a collection of Process Stages joined by workflow.

For an expanded definition please refer to page 18.

Process properties:

- **Default Execution Level** - Identifies whether the Roles assigned to the **Who can start the Process** folder can (**No one except assigned Roles**) or cannot (**Everyone except assigned Roles**) initiate the Process.
- **Stage Summary for To-do List** - Used to provide distinguishing information on the Stage that appears in the first column of the To Do List - refer to page 138 for more information.
- **Reference** - A reference caption is a user-defined labeling system that is displayed on Stages and Tasks in Process Flow diagrams. The number entered at this level is added to the beginning of the reference caption for all Stages and Tasks used in this Process. This needs to be enabled at the model-level - refer to **Reference Caption Position** on page 34 for more information.



- **Auto repeating process?** - Indicates that this Process will automatically repeat in *XSOL Automation*.



This only applies to the first Worksheet Task of the first Stage in the Process that the Role who started the Process can use. The Role who started the Process must only be able to use one Stage in the entire Process for this to work in *XSOL Automation*.

- **Archived after** - Removes all Stage and event information from the *XSOL Automation* database to the archive database after a certain time period. Enter the required number in the field followed by **d** (day), **w** (week), **m** (month), or **y** (year). Archived Processes are still visible in Process Summaries and can be restored using *XSOL Automation*.
- **Removed after** - Removes the Process and all Stage and event information from the *XSOL Automation* database to the archive database after a certain time period. Enter the required number in the field followed by **d** (day), **w** (week), **m** (month), or **y** (year). Removed Processes are not visible in Process Summaries but can be restored using *XSOL Automation*.
- **Permanently deleted after** - Permanently removes the Process and all Stage and event information from the *XSOL Automation* database and the archive database after a certain time period. Enter the required number in the field followed by **d** (day), **w** (week), **m** (month), or **y** (year). Deleted Processes cannot be restored using *XSOL Automation*.

To add a Process:

1. Click on the  **Processes** designer button in the **Design for Mapping** window.
2. Click on the  button.
3. Enter the name of the Process and press the **Enter** key to create another item or press the **Tab** key to only create this item.
4. Add or change any properties in the **Properties Editor**, if required.

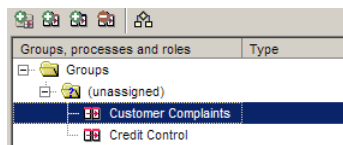



Figure 4.24: Adding a Process

5. You may now need to use the following functionality:

- The **Who can start the Process** folder enables you to indicate who can start this Process. Drag and drop any Roles that are required for this folder from the  **Roles** area in the right pane to the folder. You can also create Roles directly inside this folder - refer to page 102 for more information.



If no Roles are assigned to this folder, the Process will not be able to be initiated from the Can Do List in *XSOL Automation*.

- Stages (refer to page 54 for more information).
- Roles (refer to page 102 for more information).
- Escalation Levels (refer to page 56 for more information).

Defining Variables that map to an initiated Process



Automation Systems Design only

Once you have created your Process and the Variables required in that Process for an *XSOL Automation* system (refer to page 128 for more information), you need to indicate which Variables can be used by an "Initiate Process" Task and which Variables are mandatory, if any. This list is then used when you use an "Initiate Process" Task (refer to page 66 for more information).



You only need to define Variables for Processes that will be started by an "Initiate Process" Task. If the Process is never going to be initiated, you do not need to define this Variable list.

To define Variables that map to an initiated Process:

1. Click on the Variables Used designer button in the **Design for Mapping** window.
2. Select the required Process or Companion Process in the left pane.
3. Drag and drop any required Variables from the **Simple variables** and/or **Reference variables** folders to the **Variables to map to when initiated** folder.

All Variables are automatically mandatory. If a Variable is not mandatory, remove the tick from the tick box in the **Mandatory** column next to the required Variable.

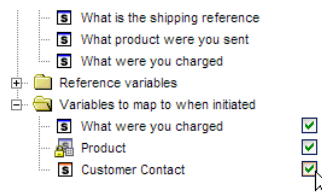


Figure 4.25: Defining Variables that map to an initiated Process

4. If you want to define all Variables for a Process, click on the **Simple variables** or **Reference variables** folder and click on the button.

If you want to remove all Variables from a Process, click on the button.

Changing a Process

Changing a Process involves adding or changing any properties for this item.

To change a Process:

1. Click on the Processes designer button in the **Design for Mapping** window.
2. Click on the required item in the list.
3. Change any properties in the **Properties Editor**, if required.

4.7.2 Using Companion Processes




Automation Systems Design only



Companion Processes are Processes that work in conjunction with their parent Process and can change information in existing Process instances. Companion Processes are used on-demand, rather than being

scheduled as part of the Parent process, and are initiated from any model displaying information from the parent Process or from the To Do List for Stages to which a Companion Process is assigned. Any number of Companion Processes can be assigned to a single parent Process and each Companion Process can have any number of Initiation Rules assigned. Each Companion Process can optionally be flagged as usable on all completed Processes, all incomplete Processes, or both.

Companion Process properties:

- **Stage Summary for To-do List** - Refer to page 54 for more information.
- **Reference** - Refer to page 54 for more information.
- **Can use on completed processes** - Can start the Companion Process if the parent Process has finished.
-  When the **Can use on completed processes** and/or **Can use on incomplete processes** options are selected one Initiation Rule must be assigned for either of both of the selections.
- **Can use on incomplete processes** - Can start the Companion Process if the parent Process is in progress.

To add a Companion Process:

1. Click on the  **Processes** designer button in the **Design for Mapping** window.
2. Click on the Process to which you want to add a Companion Process.
3. Click on the  button.
4. Enter the name of the Companion Process and press the **Enter** key to create another item or press the **Tab** key to only create this item.
5. Add or change any properties in the **Properties Editor**, if required.

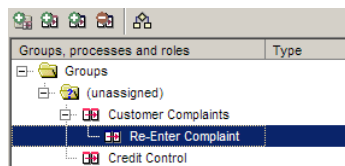


Figure 4.26: Adding a Companion Process

6. You may now need to use the following functionality:
 - Initiation rules (refer to page 53 for more information).
 - Stages (refer to page 54 for more information).
 - Roles (refer to page 102 for more information).
 - Escalation Levels (refer to page 56 for more information).

Sharing Variables from a Companion Process to a Process



Automation Systems Design only

To update an existing Process using a Companion Process in *XSOL Automation* you need to share one or more Variables from the Companion Process to the Process. These shared Variables are used by the Process Update Task (refer to page 60 for more information).

To share Variables from a Companion Process to a Process:

1. Click on the Variables Used designer button in the **Design Business Process** window.
2. Click on the required Companion Process in the left pane.
3. Tick the Variables in the Shared column in the middle pane that you are using to update the parent Process.

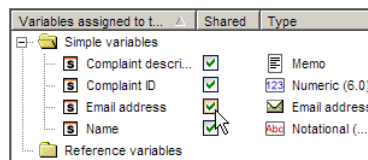


Figure 4.27: Sharing Variables from a Companion Process to a Process

Adding Variables to support external initiation



Automation Systems Design only

Companion Processes can be initiated by external applications. You can add Variables to a Companion Process which will accept external information from parameters of the same name.

To add a Variable to support external Companion Process initiation:

1. Click on the Variables Used designer button in the **Design Business Process** window.
2. Click on the required Companion Process in the left pane.
3. Drag and drop all required Variables from the Variables tab in the right pane to the **Variables to map to when externally initiated** folder in the middle pane.

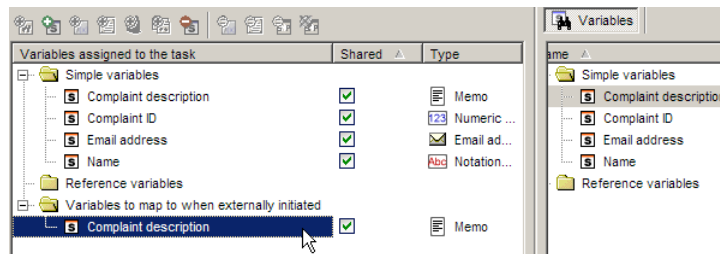


Figure 4.28: Adding a Variable to support external Companion Process initiation


Changing a Companion Process



Automation Systems Design only

Changing a Companion Process involves adding or changing any properties for this item.

To change a Companion Process:

1. Click on the  **Processes** designer button in the **Design for Mapping** window.
2. Click on the required item in the list.
3. Change any properties in the **Properties Editor**, if required.

Using Initiation Rules



Automation Systems Design only

Initiation Rules are used by Companion Processes to determine who can start a Companion Process. Companion Processes can be made available for all complete Processes, all incomplete Processes, or assigned to a Stage in the Process (useful if you wish to start a Companion Process in different Stages in the parent Process, and each Stage can be processed by different Roles, Teams, and/or users).





Initiation Rules assigned to all completed and/or all incomplete Processes cannot be assigned to Stages within the parent Process.

Initiation Rule properties:

- **Default execution level** - Determines what access all users have to this Initiation Rule. Choose one of the following options:
 - Everyone except assigned Roles
 - No one except assigned Roles
- **Override with** - Specify the user, Role and/or Team that will undertake this Stage if they have Role-based access. Refer to page 56 for more information about this property.
- **Can use on completed processes** - Can only be used on completed Processes and is only available from Process Summaries.
- **Can use on incomplete processes** - Can only be used on incomplete Processes and is only available from Process Summaries.

To add an Initiation Rule:

1. Click on the Companion Process to which you want to add an Initiation Rule in the **Design for Mapping** window.
2. Click on the  button.
3. Enter the name of the Initiation Rule and press the **Enter** key to create another item or press the **Tab** key to only create this item.
4. Drag and drop any Roles that are required for this Initiation Rule from the  **Roles** area in the right pane to the Initiation Rule. You can also create Roles directly inside an Initiation Rule - refer to page 102 for more information.

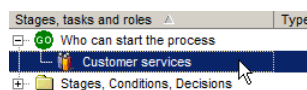



Figure 4.29: Adding an Initiation Rule

5. Add or change any properties in the **Properties Editor**, if required. Go to the next step for more specific information about some of these properties.
6. If you want to use this Initiation Rule for completed Processes, select the **Can use on completed processes** tick box in the **Properties Editor**. Only one Initiation Rule can have this property selected per Companion Process.

If you want to use this Initiation Rule for incomplete Processes, select the **Can use on incomplete processes** tick box in the **Properties Editor**. Only one Initiation Rule can have this property selected per Companion Process.

If you want to use this Initiation Rule in another way in the Process, please follow these steps:

- (a) Select the option on how the Companion Process will be used from the **Override with** property (refer to page 56 for more information).
- (b) If the **Select Stage** property is now available, select the required Stage from the drop-down list.
- (c) Click on the parent Process in the left pane, then the  **Companion Processes** tab in the right pane.
- (d) Drag and drop the Companion Process whose Initiation Rules you are working on from the right pane to the **Companion Processes** folder underneath the required Stage in the middle pane.
- (e) Select the **Use this rule** tick box for the Initiation Rule that you want to use for this Stage.

Changing an Initiation Rule



Automation Systems Design only

Changing an Initiation Rule involves adding or changing any properties or Roles for this item.

To change an Initiation Rule:

1. Click on the required Companion Process in the **Design for Mapping** window.
2. Click on the required Initiation Rule in the middle pane.
3. Change any properties in the **Properties Editor**, if required.

4.7.3 Using Stages

A Stage defines a step that comprises one or more Tasks, which can be scheduled within a Process. If there are one or more Stages within a Process you can design Stage Process Flow for those Stages (refer to page 80 for more information).

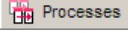

For an expanded definition please refer to page 18 for more information.

Stage properties:

- **Stage Summary for To-do List** - Used to provide distinguishing information for the Stage that appears in the first columns of the To Do List - refer to page 138 for more information.
- **Schedule Expression** - Used to schedule when the next Stage is required to be completed by - refer to page 138 for more information.
- **Reference** - A reference caption is a user-defined labeling system that is displayed on Stages and Tasks in Process Flow diagrams. The number entered at this level is added after the Stage caption for all Stages and Tasks used within this Process. This needs to be enabled at the model-level - refer to **Reference Caption Position** on page 34 for more information.

- **To-do Values Action** - Allows you to choose whether Process Variables in the To Do List appear before, after, or instead of Stage Process Variables in a run-time system. Only available at the Stage level.
- **Performed by System** - If ticked the Stage can only be executed by the *XSOL Automation* system, not a user, and is displayed as 🤖. System Stages cannot have any Escalation Levels and can only use Tasks that *XSOL Automation* can perform, such as **System Calculation**, **Document Operation**, or **Send Email** Tasks. This property is not visible if the Stage contains one or more Tasks that *XSOL Automation* cannot perform or if any Roles have been assigned.
- **Allow stage confirmation bypass** - If ticked this Stage does not display a list of completed Tasks after the Stage has been successfully processed in *XSOL Automation*. This property is not available if the **Performed by System** option is selected.
- **Allow stage to be manually parked?** - If ticked this Stage can be parked at any time by a user in *XSOL Automation*. A parked Stage is paused - all Tasks that have already been completed, and any entered data, is saved, and the Stage can be restarted at any time.

To add a Stage:

1. Click on the  designer button in the **Design for Mapping** window.
2. Click on the  button.
3. Enter the name of the Stage and press the **Enter** key to create another item or press the **Tab** key to only create this item.
4. Change the selected **Can be executed by** drop-down list option, if required.
5. Add or change any properties in the **Properties Editor**, if required.

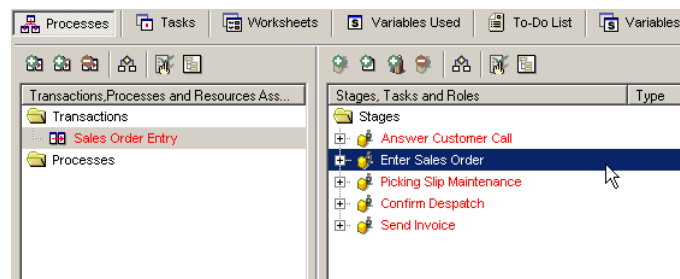
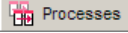


Figure 4.30: Adding a Stage

Changing a Stage

Changing a Stage involves adding or changing any properties for this item.

To change a Stage:

1. Click on the  designer button in the **Design for Mapping** window.
2. Click on the required Stage.
3. Change any properties in the **Properties Editor**, if required.

4.7.4 Using Escalation Levels




Escalation Levels are a means of assigning Roles to any scheduled Stages in an *XSOL Automation* system after a set period of time. This moves overdue Stages onto the To Do Lists of managers or other employees and allows them to be actioned even if the first set of assigned users cannot access the system. Escalation Levels control which Roles or users can use specific Processes.

You can also control what happens if the last Escalation Level is not actioned by sending the Process Flow to another Stage to handle the exception - refer to page 57 for more information.

Escalation Level properties:

- **Default access level** - Choose one of the following options:
 - Everyone can execute this Stage
 - Everyone can see this Stage on the To Do List
 - Role assignments determine access to this Stage
- **Override with** - Specify the user, Role and/or Team that will undertake this Stage if they have Role-based access. Choose one of the following options:
 - **(no override)** - Assigned Roles determine who has access to this Stage.
 - **same role & team as stage** - Only the Roles in the Team that performed the Stage selected in the **Select Stage for Assignee** drop-down list can undertake this Stage.
 - **role & team that started process** - Only the Role and Team that initiated this Process can undertake this Stage.
 - **same role as stage** - Only the Role that performed the Stage selected in the **Select Stage for Assignee** drop-down list can undertake this Stage.
 - **role that started process** - Only the Role that initiated this Process can undertake this Stage.
 - **same team as stage** - Only the Team that performed the Stage selected in the **Select Stage for Assignee** drop-down list can undertake this Stage.
 - **team that started process** - Only the Team that initiated this Process can undertake this Stage.
 - **same user as stage** - Only the user who performed the Stage selected in the **Select Stage for Assignee** drop-down list can undertake this Stage.
 - **user that started process** - Only the user who initiated this Process can undertake this Stage.
 - **specified user** - Only the indicated user can undertake this Stage. From the **Specify User** drop-down list select the **Input the User** option to manually enter the user in the **User** field or select the **Defined by a user type variable** field and select the Variable from the **Get user from** drop-down list.
- **Negate the override?** - If ticked reverses the meaning of the selected **Override with** option.
- **Timeout** - Indicates when this Escalation Level finishes and the Process Flow escalates to another Escalation Level or goes to another Stage. Choose one of the following options:
 - **Input a Value** - Enter the required number in the **Timeout Value** field followed by **m** (minute), **h** (hour), **d** (day), **w** (week), **mo** (month), or **y** (year).
 - **Use a calculation** - Open the **Expression Editor** for this field and enter an Expression to calculate the timeout value.

To add an Escalation Level:

1. Click on the  **Processes** designer button in the **Design for Mapping** window.
2. Click on the required Stage for the Process.
3. One Escalation Level is automatically created when a Stage is created. If you need to create a new Escalation Level, click on the  button and enter the name of the Escalation Level.
4. Click on the preceding Escalation Level to the one you just created. If you did not create an Escalation Level, click on the first one.
5. Change any properties in the **Default access level** or **Override with** fields, if required. You can reverse the meaning of most selected options in the **Override with** drop-down list by clicking on the **Negate assignment** tick box.
6. Drag and drop any Roles that are required for this Escalation Level from the  **Roles** area in the right pane to the Escalation Level. You can also create Roles directly inside an Escalation Level - refer to page 102 for more information.
7. If you do not have a following Escalation Level, you are finished with this procedure.
If you do have a following Escalation Level, you need to enter a time calculation or fixed time period after which this Stage escalates to other user's To Do Lists.
8. If you need to enter a fixed time period, select the **Input a Value** option from the **Timeout** field in the **Properties Editor**. Enter the required number in the **Timeout Value** field followed by **m** (minute), **h** (hour), **d** (day), **w** (week), **mo** (month), or **y** (year).

Property	Value
Caption	Escalate to HR Manager
Default access level	Role assignments determine access...
Assignee for this stage	Based on role
Timeout	Input a Value
Timeout Value	2 days
Runtime Hint	
Explanation	
Version Note	
Designer comments	
Help URL	
External ID	Escalate_to_HR_Manager

Figure 4.31: Setting the timeout required to activate the next Escalation Level

If you need to enter a time calculation, select the **Use a calculation** option from the **Timeout** field in the **Properties Editor**. Double-click on the **Calculation for Timeout** field and enter an Expression. Refer to page 138 if you need more information about creating an Expression.

9. Add or change any properties in the **Properties Editor**, if required.



If the last Escalation Level is not actioned, you can send the Process Flow to another Stage to handle this exception. You need to add a timeout value to the last Escalation Level and then link the Stage to the exception-handling Stage in the **Design Process Flow** window (refer to page 94 for more information).

Using exception flow

Exception flow is used to deal with a Stage that has not been actioned, even if Escalation Levels have been used. This allows you to plan for an exception to the standard Process Flow.

To use exception flow:

1. Add a value or calculation to the **Timeout** property of the last Escalation Level for a Stage. Refer to page 56 for more information about using Escalation Levels.
2. You can now link the Stage to another Stage in the Stage Process Flow diagram. You can change the link text by clicking on it and changing the Caption property in the **Properties Editor**.

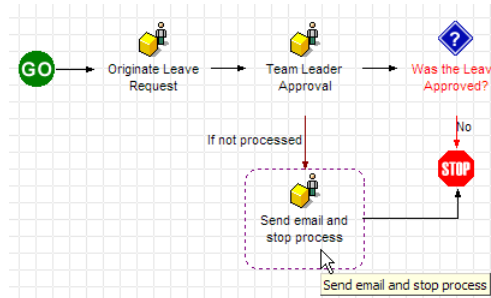


Figure 4.32: Using exception flow

Changing an Escalation Level

Changing an Escalation Level can involve changing the security settings for the Escalation Level, adding or removing Roles, and altering the timeout values to activate the next Escalation Level.

To change an Escalation Level:

1. Click on the required Escalation Level for the required Process and Stage in the **Design for Mapping** window.
2. Add or remove Roles from the Escalation Level, if required.
3. Alter the properties for the Escalation Level, if required.

Caption	Escalate to HR Manager
Default access level	Role assignments determine access...
Assignee for this stage	The same role and team as the sele...
Negate stage assignee	<input type="checkbox"/>
Select Stage for Assignee	
Timeout	Input a Value
Timeout Value	1 day
Runtime Hint	
Version Note	
Designer comments	
Help URL	
External ID	Escalate_to_HR_Manager
(Debug) Entity ID	uuid_0FDB9372-16FD-BA9C-476C-...

Figure 4.33: Changing an Escalation Level

4.7.5 Designing a To Do List display



Automation Systems Design only

Models created by the *XSOL Mapping* can be used to create a running system using *XSOL Automation*. Each Stage in a Process will appear on a user's To Do List at some time after the Process has been instigated. You can set what columns are displayed and in what order they are sorted, based on the Variables that are available for the Stage.

You can also create an Expression that displays a summary of Stage information in the first column of the To Do List - refer to page 138 for more information.



You can not use Work Variables in a To Do List entry.



To Do List properties:

- **To-do Values Action** - Allows you to choose whether Process Variables in the To Do List appear before, after, or instead of Stage Process Variables in a run-time system. Only available at the Stage level.

Variable properties:

- **Inherit Label?** - If ticked the Variable (as a column heading) uses the same caption as the source Variable. Since long Variable names can take up a lot of screen real-estate in *XSOL Automation* it is a good idea to shorten Variable labels.
- **Default Width** - Enter the default column size, in millimeters, for this Variable.

To design a To Do List display:

1. Click on the  **To-Do Lists** designer button in the **Design for Mapping** window.
2. Click on the required Process or Stage in the available list.
3. Drag and drop as many Variables as required from the  **Variables** area in the right pane into the **Process Variables Selected** or **Stage Variables Selected** folder. Each folder indicates whether the Variables information comes from the Process or Stage level. The order of the Variables, from top to bottom, dictates how the columns in the To Do List appear, from left to right. You can re-arrange the order of items by clicking on a Variable and dragging it to the required position.

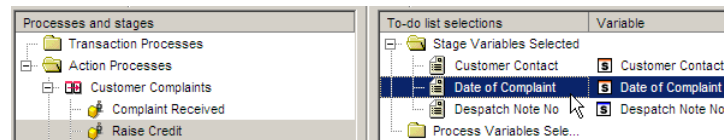





Figure 4.34: Designing a To Do List display

4. To set the sort order of the displayed information in the run-time system, click on the  button.
5. Drag and drop the listed Variables to the required location in the list, or click on the required Variable and click on the  or  button. The order of the Variables, from top to bottom, dictates which column is sorted first.
6. For each Variable that you want sorted, select the required sort direction from the **Sort Direction** column drop-down list. Select one of the following options:
 - **Ascending** - The column is sorted in ascending sequence - alphabetical order, 1-10.
 - **Descending** - The column is sorted in descending sequence - reverse-alphabetical order, 10-1.
 - **None** - The column is not sorted in any specific order.

- Repeat for as many Variables as required, and click on the **OK** button when you have finished.

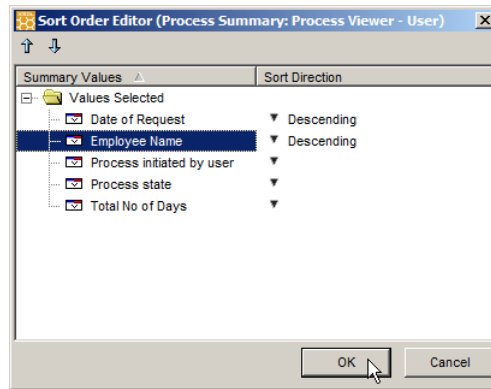


Figure 4.35: Changing the sort order of To Do List Variables

- Click on the **Yes** button to confirm any changes that you have made.

Changing a To Do List display



Automation Systems Design only

Changing a To Do List display involves altering the Variables displayed in the To Do List and changing any properties that you have added to the Variables in the To Do List.

To change a To Do List display:

- Click on the **To-Do Lists** designer button in the **Design for Mapping** window.
- Click on the required Process or Stage in the available list.
- If you need to re-arrange the order of items, click on a Variable and drag it to the required position.
If you need to set the sort order of any of the Variables as they appear in the run-time system, click on the button.
If you need to delete a Variable, click on the required item and click on the **Delete** key.

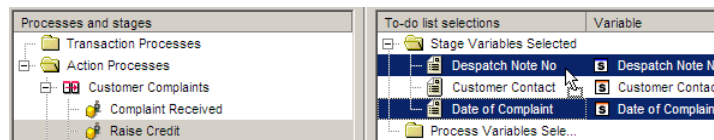


Figure 4.36: Changing a To Do List display

4.7.6 Using Tasks



A Task is an activity undertaken within a Stage that is not scheduled but rather initiated when any previous Task within the Stage is completed. Various types of Tasks can be represented including worksheets, reports, launching external applications (Excel, Word, ERP (Enterprise Resource Planning) program, etc.), manual operations, document operations, updating of Reference Data, sending emails, and so on. Tasks can be

reused throughout an Enterprise or Trading Entity. A Task assigned to a Stage is known as a Task instance and the Task it is assigned from is known as the parent Task.

If there are one or more Tasks within a Stage you can design a Task Process Flow for those Tasks (refer to page 80 for more information). Tasks can be reused between different Processes, as required.

The **Properties Editor** for a Task instance displays both the properties for the parent Task and the Task instance. The parent Task properties are displayed at the top of the **Properties Editor** under the **Task** label and are in blue. All the other properties are for the Task instance.








Both the parent Task and the Task instance can have User Notes. If you are in the **Design for Mapping** window and you see the  icon next to the Task in the middle pane, that means the Task instance has one or more User Notes. If the you see the  icon next to the Task in the right pane, that means the Task parent has one or more User Notes.

The following Task types are available:

-  **Create Document from a Template** - Generate a Microsoft Word 2003 or Microsoft Excel 2003 document using a template (refer to page 69 for more information). The following additional fields are available:
 - **Template Document** - Select the document to generate.
If you need to use a Variable, select the **Defined by a document type variable** option and select the required Variable from the **Get document name from variable** drop-down list.
If you need to use a Document Location, select the **Defined by a document location and file path** option, select the required Document Location from the **Use Document Location** drop-down list and enter the relative path to the document in the **And Relative File Path** field.
 - **Map to Template Variables** - Select the Microsoft Office 2003 template whose bookmarks you want to map to Variables.
-  Only Microsoft 2003 or Microsoft Excel 2003 templates in XML form are supported.
- **Document to Process** - Select the document name that will be generated.
If you need to use a Variable, select the **Defined by a document type variable** option and select the required Variable from the **Get document name from variable** drop-down list.
If you need to use a Document Location, select the **Defined by a document location and file path** option, select the required Document Location from the **Use Document Location** drop-down list and enter the relative path to the document in the **And Relative File Path** field.
-  **Document Operation** - Allows users to copy , delete , edit , move , or view  a specific document. The following additional fields are available (there vary depending on which operation you choose):
 - **Copy/Move Document from/to** - Select the document to copy or move. Both the **from** and **to** fields have the same behaviour.
If you need to use a Variable, select the **Defined by a document type variable** option and select the required Variable from the **Get document name from variable** drop-down list.
If you need to use a Document Location, select the **Defined by a document location and file path** option, select the required Document Location from the **Use Document Location** drop-down list and enter the relative path to the document in the **And Relative File Path** field.
 - **Document to Delete/Edit/View** - Select the document to delete, edit, or view.
If you need to use a Variable, select the **Defined by a document type variable** option and select the required Variable from the **Get document name from variable** drop-down list.






If you need to use a Document Location, select the **Defined by a document location and file path** option, select the required Document Location from the **Use Document Location** drop-down list and enter the relative path to the document in the **And Relative File Path** field.




-  **Initiate process** - The task initiates one of the other Processes in the system. The following additional fields are available:
 - **Initiated process** - Click on the required Process that the Task initiates from the drop-down list.
 - **Design tasks - process variables map** - Opens a window that allows you to map existing Variables (from the right pane) onto Variables used in the initiated Process (left pane). Refer to page 66 for more information.
-  **Manual Task** - The user needs to perform a manual task outside of the system. The following additional fields are available:
 - **Confirmation Required** - Select if the user needs to indicate that the Manual Task has been performed. This is displayed as .
 - **Message** - Enter any text that you want the *XSOL Automation* user to see. You can enter a maximum of 4000 characters.
 If you need to use a Variable, select the **Get from a variable** option and select the required Variable from the **Get Message From** drop-down list.
 If you need to use plain text, select the **Input the message** option and enter the text in the **Manual Message** field.
-  **Non-XSOL** - Opens a Non-XSOL task. The following additional fields are available:
 - **Target** - Enter the path to the application that is opened.
 If you need to use a Variable, select the **Get from a variable** option and select the required Variable from the **Get Target From** drop-down list.
 If you need to use plain text, select the **Input the target** option and enter the text in the **Target** field.
 - **Open Target As** - The following options control how a Non-XSOL task is handled by the system.
 - * **Automatic** - Automatically launches the target outside of the system when the Task Worksheet is opened.
 - * **Embedded** - Automatically launches the target inside the system when the Task Worksheet is opened.
 - * **Manual** - Provides a link to the target within the Task Worksheet that can optionally be activated by the user.
 - **Allow Relaunch** - Select if the user is allowed to launch the target multiple times. If not selected, the user is only allowed to launch the target once.
 - **Icon** - Double-click on the field and click on the display icon for this Task. Refer to page 173 for more information about adding new Task icons.
-  **Print Document** - Print a document.









The XPM machine must have the application installed on it that opens the type of document you want to print. For example, to print a Microsoft Word document the XPM machine needs to have Microsoft Word installed.

The following additional fields are available:

- **Document to Print** - Select the document to print.
If you need to use a Variable, select the **Defined by a document type variable** option and select the required Variable from the **Get document name from variable** drop-down list.
If you need to use a Document Location, select the **Defined by a document location and file path** option, select the required Document Location from the **Use Document Location** drop-down list and enter the relative path to the document in the **And Relative File Path** field.
- **Printer to use** - Select the printer to use.
If you need to use a Variable, select the **Defined by a printer type variable** option and select the required Variable from the **Get Printer From** drop-down list.
If you need to use a Logical Printer, select the **Defined by a logical printer** option and select the required Logical Printer from the **Use Logical Printer** drop-down list.
-  **Process Update** - The task updates one of the other Processes in the system (refer to page 52). The following additional fields are available:
 - **Process to Update** - Click on the required Process that the Task updates from the drop-down list.
 - **Variables to Update to Parent Process** - Opens a window that allows you to map existing Variables (from the right pane) onto Variables used in the updated Process (left pane). Refer to page 66 for more information.
-  **Script** - The task executes a Python, VB, or Javascript Expression. This Expression can do anything that these scripting languages can perform, and is often used to pass information to and from an external application's API (Application Program Interface) or database. Script Task Expressions have access to all Variables and can manipulate the Variable's source XML. Refer to page 71 and page 221 for more information on Script Tasks and how to use them.
The following additional fields are available:
 - **Execute on Server** - Tick to perform this Expression on the *XSOL Automation* server machine, and is displayed as . If not ticked, the Expression is performed on the *XSOL Automation* client machine, and is displayed as .
 - **Script Language** - Select the required scripting language.
-  **Send Email** - Send an email. The following additional fields are available:
 - **To** - Double-click on the field. You can now assign email addresses to the **To** field by entering the address in the top field or dragging and dropping Variables or Roles onto the **To** area.
 - **Cc** - Same as the **To** property.
 - **Bcc** - Same as the **To** property.
 - **From** - The sender's email address.
If you need to use a Variable, select the **Defined by an email address type variable** option and select the required Variable from the **Get From Address From** drop-down list.
If you need to manually enter the address, click on the **Input the email address** option and enter the address in the **From Address** field.
 - **Reply To** - The email address that is used when replying to the email.
If you need to use a Variable, select the **Defined by an email address type variable** option and select the required Variable from the **Get Reply To Address From** drop-down list.
If you need to manually enter the address, click on the **Input the email address** option and enter the address in the **Reply To Address** field.

- **Subject Source** - The subject of the email.
If you need to use a Variable, select the **Get from a variable** option and select the required Variable from the **Get Subject From** drop-down list.
If you need to manually enter the subject, click on the **Input the subject** option and enter the address in the **Subject** field.
- **Body Source** - The main text of the email. You can enter a maximum of 4000 characters.
If you need to use a Variable, select the **Get from a variable** option and select the required Variable from the **Get Body Text From** drop-down list.
If you need to manually enter the body text, click on the **Input the body text** option and enter the address in the **Body Text** field.
-  **Send Fax** - Send a fax. The following additional fields are available:
 - **Document to Send** - Select the document to fax.
If you need to use a Variable, select the **Defined by a document type variable** option and select the required Variable from the **Get document name from variable** drop-down list.
If you need to use a Document Location, select the **Defined by a document location and file path** option, select the required Document Location from the **Use Document Location** drop-down list and enter the relative path to the document in the **And Relative File Path** field.
 - **Phone No to Send To** - Select the phone number to use.
If you need to use a Variable, select the **Defined by a phone no type variable** option and select the required Variable from the **Get Phone No From** drop-down list.
If you need to manually enter the phone number, select the **Input the phone number** option and enter the required phone number in the **Phone Number** field.
-  **Send Pager Message** - Send a pager message. The following additional fields are available:
 - **Message Source** - Select the pager information.
If you need to use a Variable, select the **Get from a variable** option and select the required Variable from the **Get Message From** drop-down list.
If you need to manually enter the message, select the **Input the Message** option and enter the pager message in the **Message** field.
 - **Phone No to Send To** - Select the phone number to use.
If you need to use a Variable, select the **Defined by a phone no type variable** option and select the required Variable from the **Get Phone No From** drop-down list.
If you need to manually enter the phone number, select the **Input the phone number** option and enter the required phone number in the **Phone Number** field.
-  **Send SMS Message** - Send a SMS message. The following additional fields are available:
 - **Message Source** - Select the SMS message information.
If you need to use a Variable, select the **Get from a variable** option and select the required Variable from the **Get Message From** drop-down list.
If you need to manually enter the message, select the **Input the Message** option and enter the pager message in the **Message** field.
 - **Phone No to Send To** - Select the phone number to use.
If you need to use a Variable, select the **Defined by a phone no type variable** option and select the required Variable from the **Get Phone No From** drop-down list.
If you need to manually enter the phone number, select the **Input the phone number** option and enter the required phone number in the **Phone Number** field.

-  **System Calculation** - Automatically process all of the Variables and Python scripts that are attached to the Task. Also known as a System Task.
-  **Table Update** - Allows users to delete , insert , or update  records in a Table (refer to page 112 for more information). The following additional fields are available:
 - **Process** - Choose whether you need to delete, insert, or update a datafield in a Table.
 - **Reference data variable** - Click on the required Table to delete, update, or insert from the drop-down list.
 - **Design data fields - process variable map** - Choose the required datafield(s) that map to the Table that will be updated - refer to page 66 for more information on Variable mapping.
-  **Worksheet** - Accepts user input using a specific Worksheet (refer to page 105 for more information). The following additional fields are all associated with Worksheet design (refer to page 105 for more information):
 - **Default Label Alignment** - Click on the default label alignment for labels on any Worksheets associated with this Task from the drop-down list.
 - **Default Font Name** - Click on the default font used on any Worksheets associated with this Task from the drop-down list.
 - **Default Font Size in Pts** - Click on the default font size used on any Worksheets associated with this Task from the drop-down list.



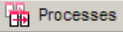

Tasks types can only be changed at the Task list level (in the **Design for Mapping** window).

For an expanded definition please refer to page 18 for more information.

Task properties:

- **Reference** - A reference caption is a user-defined labeling system that is displayed on Stages and Tasks in Process Flow diagrams. The number entered at this level is added to the after the Process reference caption and after the Stage caption for all Stages and Tasks used in this Process. This needs to be enabled at the model-level - refer to page 100 for more information.

To add a Task:

1. Click on the  designer button in the **Design for Mapping** window.
2. Click on the  button.
3. Enter the name of the Task and press the **Enter** key to create another item or press the **Tab** key to only create this item.
4. Click on the required option from the **Task Type** drop-down list in the **Properties Editor**. Depending on which option you selected you may need to select further information from more fields in the **Properties Editor**.
5. Add or change any properties in the **Properties Editor**, if required.

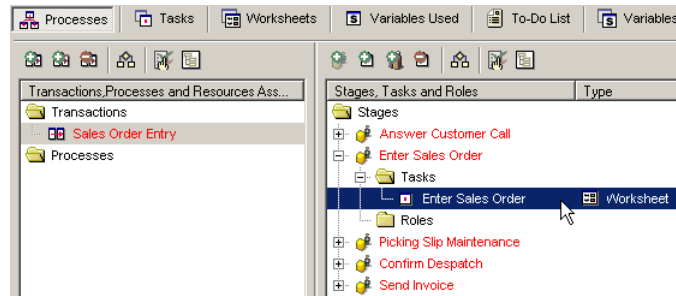


Figure 4.37: Adding a Task

Mapping a Variable from one Process to another



Automation Systems Design only

You can start a Process from another Process by using an "Initiate Process" Task. When you initiate another Process in an *XSOL Automation* system you need to associate Variables from the first Process across to the second Process so that they can be used.



The list of available Variables that can be mapped is defined when you create a Process (refer to page 50 for more information).

To map Variables from one Process to another:

1. Click on the **Processes** designer button in the **Design for Mapping** window.
2. Click on the required "Initiate Process" Task. Make sure that a Process has been selected in the **Initiated process** property in the **Properties Editor**.
3. Click on the **Design task - process variables map** property in the **Properties Editor** and click on the button.
4. To manually map Variables:
 - (a) Drag and drop Variables from the right pane to the left pane.
 - (b) Select the Variables that you want to map from in the **Task Variables** column for each Variable. You must select a Variable to map from if the **Mandatory** tick box is ticked.

To automatically map Variables from a specific Process:

- (a) Select the Process from the **Auto map variables using process** drop-down list and click on the button.

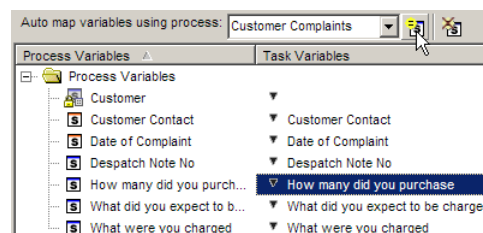


Figure 4.38: Automatically mapping Variables from one Process to another

You must select a Variable to map from if the **Mandatory** tick box is ticked.





To remove the mapping on a Variable, select the Variables and click on the  button.

Adding an attachment to a Send Email Task

You can add any number of attachments to a **Send Email** Task.

To add an attachment to a Send Email Task:

1. Click on the  designer button in the **Design for Mapping** window.
2. Click on the required Send Email Task.
3. Click on the  button.
4. Select the required file using **Properties Editor**.

If you need to use a Variable, select the **Defined by a document type variable** option and select the required Variable from the **Get document name from variable** drop-down list.

If you need to use a Document Location, select the **Defined by a document location and file path** option, select the required Document Location from the **Use Document Location** drop-down list and enter the relative path to the document in the **And Relative File Path** field.

Property	Value
Document to Attach	Defined by a document location...
Use Document Location	Security Documents
And Relative File Path	Administration.doc

Figure 4.39: Adding an attachment to a Send Email Task

Selecting a document in XSOL Automation



Automation Systems Design only

You need to use a Business Rule if you need to design your model so that a specific document can be selected via a Worksheet in *XSOL Automation*.

The following document selection options are available for Business Rules:

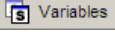



- **Should Document exist** - Select one of the options to determine whether the indicated document exists, does not exist, or it does not matter.
- **Select the Document** - If ticked a document can be selected and does not have to be specified in the Document Variable.
- **Edit the Document** - If ticked the indicated document can be edited. If not ticked, the document can only be viewed.
- **Can Add to Location** - If ticked this allows Users to add new documents to the Document Store - the selected document will automatically be copied to the Document Store location. The **Select the Document** tick box must be ticked as well for this to work.

The following document selection options are available for Locations:

- **Document Location** - Select the Document Location from which documents are chosen.

- **Source Document** - Select the document name from the following options:
 - **None** - A document name is not used at this point. A document can still be selected, depending on the selected properties for the Business Rule.
 - **From variable** - Select the required Document Variable from the **Get Document From** drop-down list.
 - **Specify value** - Enter the document path, relative to the Document Location, in the **Document name and path** field. You need to enter the folder path with the following syntax: `/folder/subfolder`

To select a document in XSOL Automation:

1. Click on the  designer button in the **Design for Automation** window.
2. Select the required **Document** Variable from the Variable list, or create a new one (refer to page 128 for more information).
3. Click on the  button to create a new Business Rule for the Variable. Rename the Business Rule, if required.
4. Change the properties for the Business Rule as described in the description of this topic, if required.
5. If you do not specify a Document Location then all Document Locations are available. To restrict this Business Rule to one or more Document Locations:
 - (a) Click on the  button in the middle pane.
 - (b) Change the properties for the Location as described in the description of this topic, if required.
6. If you need to set filters on the required document types that a user can select, click on the **Select Document Types** property and click on the  button.
7. You can set filters in several different ways. You can use some or all of these methods, if required.
 - If you need to add a predefined filter, select the required item from the **Predefined filters** drop-down list and click on the **Add** button. You can add several predefined filters, if required. Refer to page 176 for more information about creating predefined filters.
 - If you need to add a filter manually, enter the description of the filter in the **Name** column and the filter itself in the **Filter** column. You can use the asterisk character (*) as a wildcard - for example, a filter to only choose Microsoft Word documents would be `*.doc`. You can create a new filter line by pressing the **down arrow** key, or remove an empty line by pressing the **up arrow** key.
 - If you need to set a default filter, select the required option from the **Default filter** drop-down list. You need to add at least one entry to the main filters area in order to set a default filter.
8. Click on the **OK** button.

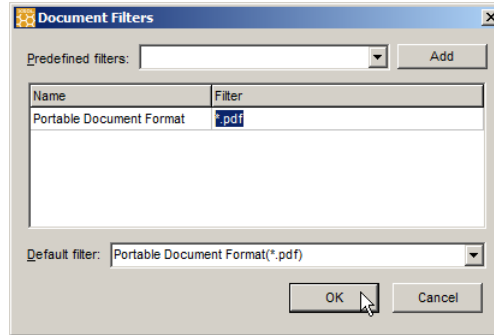


Figure 4.40: Setting predefined filters for a Document Variable

Using a Microsoft Office template with a Task



Automation Systems Design only

You can use a "Create Document from a Template" Task and a Microsoft Word or Microsoft Excel template to create Word and Excel documents automatically (including images) using Process data. This allows you to create consistent customised documents automatically within *XSOL Automation*.



You can only use Microsoft Word 2003 or Microsoft Excel 2003 documents in XML (Extensible Markup Language) format.



You need to turn off spell- and grammar-checking for Word templates, as this may cause incorrect generated documents. To do this you need to remove the tick from the **Check spelling as you type** and **Check grammar as you type** tick boxes in the **Tools > Options... > Spelling & Grammar** tab.

To use a Microsoft Office template with a Task:

1. Create a Microsoft Word 2003 or Microsoft Excel 2003 template.
2. Add one or more custom Document Properties in Word, or one or more named cells in Excel.



You need to use the underscore character (_) instead of spaces in the name for both Word and Excel.

To add a custom Document Property in Word:

- (a) Click on the **File > Properties** menu item.
- (b) Click on the **Custom** tab.
- (c) Enter the name of the custom Document Property in the **Name** field.
- (d) Enter the name of the Document Property in the **Value** field.
- (e) Click on the **Add** button.

To add a named cell in Excel:

- (a) Click on the cell that you want to add a name to.
- (b) Click on the **Insert > Name > Define...** menu item.
- (c) Enter the required name in the **Names in workbook** field.
- (d) Click on the **OK** button.

- (e) Enter some placeholder text, such as "0" or "text" or "image" (anything will do), in the named cell. Empty cells will not update correctly when the document is generated.
 - (f) If this cell will have an image in it, resize the height of the cell to be the required height of the image. The height of the cell is used to determine the height and width of the image that will be inserted.
3. If you are using Word, you now need to insert the custom Document Properties into the document where you want the values to appear.



If you need to insert an image, use the procedure after this one.

- (a) Click in the document where you want to insert a Document Property.
- (b) Click on the **Insert >Field...** menu option.
- (c) Click on the **DocProperty** option in the **Field names** list.
- (d) Double-click on the required custom Document Property to insert it.
- (e) Repeat for as many Document Properties as required.

If you are inserting an image, use the following procedure:

- (a) Click in the document where you want to insert a Document Property.
- (b) Click on the **Insert >Picture >From File...** menu option.
- (c) Insert the required image and re-size, if necessary.
- (d) Double-click on the image and click on the **Web** tab in the **Format Picture** dialog.
- (e) Enter the following text: **xsol picture:logo**. "xsol picture:" is the prefix so Automation can find the image, and "logo" is the name of the custom property to use.

- 4. Save the template as an XML document.
- 5. Select or create the Create Document from a Template-type Task in *XSOL Mapping*.
- 6. Select the document template to use from the **Template Document** property in the **Properties Editor**, and select the document name that will be generated from the **Document to Process** property. Use the following instructions for both properties:
 - If you need to use a Variable, select the **Defined by a document type variable** option and select the required Variable from the **Get document name from variable** drop-down list.
 - If you need to use a Document Location, select the **Defined by a document location and file path** option, select the required Document Location from the **Use Document Location** drop-down list and enter the relative path to the document in the **And Relative File Path** field. You need to use a Document Location for inserting images.
- 7. Open the **Map to Template Variables** property.
- 8. Navigate to and double-click on the required Word or Excel template file (in XML format).
- 9. To manually map Variables, drag and drop the required Variable from the right pane to the **Variables** column next to each bookmark.

To automatically map Variables to bookmarks of the same name, click on the  button.

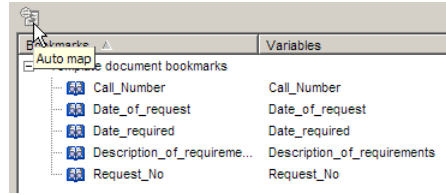


Figure 4.41: Mapping bookmarks to Variables

10. Click on the **OK** button.



To remove a document bookmark, you need to delete the use of the custom Document Properties in Word and its definition, or delete the named cell definition in Excel, then save the template and re-load it into *XSOL Mapping*.

Using the Script Task



Automation Systems Design only

The Script Task executes a Python, VB, or Javascript Expression. This Expression can do anything that these scripting languages can perform, and is often used to pass information to and from an external application's API (Application Program Interface) or database. Script Task Expressions have access to all Variables and can manipulate the Variable's source XML. You can also select whether the Variables attached to a Script Task are modifiable or not. If modifiable, the Expression can return different values than the Variable started with. Refer to page 221 for more information.

To use the Script Task:

1. Select the required Task and choose the **Script** option from the **Task Type** drop-down list in the **Properties Editor**.
2. Select the required scripting language used by the Task from the **Script Language** drop-down list.
3. Drag and drop as many Variables as required from the **Variables** area in the right pane into the **Simple variables** folder.
4. To make a Variable modifiable, select the tick box in the **Modifiable** column next to the required Variable in the middle pane. Repeat for as many Variables as required.

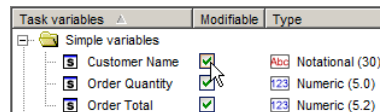


Figure 4.42: Adding Variables to a Script Task

5. Open the **Expression Editor** by double-clicking on the **Expression** property.
6. Use the `getTaskDataXml ()` function if you need to access the Task's XML structure to send to an external application. Refer to page 221 for more information.
7. To return multiple Variables:
 - (a) If you are using Python and do not want to return an XML document, simply declare the Variables with their required values:


```
Customer_Name = 'James'
Customer_ID = None
```

- (b) If you are using Javascript or VBscript, or you want to use Python to return an XML document, you need to construct an XML document using these scripting languages.

Python Example

```
# Test Python
import xsol.xml

# Create the DOM to return data
xml = xsol.xml.Xml()
xml.Xml = '<?xml version="1.0" encoding="utf-8" ?><scriptTaskResult xmlns="http://www.xsol.com,
root = xml.RootNode
namespace = 'http://www.xsol.com/scriptTaskData'

# Set variable Customer_Name = 'James'
node = xml.createNode (xsol.xml.NODE_ELEMENT, 'Customer_Name', '')
root.appendChild (node)
node.Text = 'James'

# Set variable Customer_ID = None
node = xml.createNode (xsol.xml.NODE_ELEMENT, 'Customer_ID', '')
root.appendChild (node)
node.setAttribute ('xsi:nil', 'true')

# Set variable User = 'Uid'
node = xml.createNode (xsol.xml.NODE_ELEMENT, 'User', '')
root.appendChild (node)
node.Text = '{B48B2229-D7FA-4E8A-96F7-DAFC56199AF0}'

return xml.Dom
```

VBScript Example

```
rem VBScript example
rem Create the DOM to return
dim dom, root, node
set dom = CreateObject ("Msxml2.DOMDocument.4.0")
dom.loadXML ("<?xml version=""1.0"" encoding=""utf-8"" ?><scriptTaskResult xmlns=""http://www.
set root = dom.documentElement

rem Returns variable Customer_Name = 'James'
set node = dom.createElement("Customer_Name")
root.appendChild(node)
node.text = "James"

rem Returns variable Customer_ID = Null
set node = dom.createElement("Customer_ID")
root.appendChild(node)
node.setAttribute "xsi:nil", "true"

rem Returns variable User = 'Uid'
set node = dom.createElement("User")
root.appendChild (node)
```

```
node.text = "{B48B2229-D7FA-4E8A-96F7-DAFC56199AF0}"

set Result = dom
```

Javascript Example

```
// Javascript example
// Create the DOM to return
var dom, root, namespace, node
dom = new ActiveXObject ('Msxml2.DOMDocument.4.0')
dom.loadXML ('<?xml version="1.0" encoding="utf-8" ?><scriptTaskResult xmlns="http://www.xsol.com/scriptTaskData'
root = dom.documentElement
namespace = 'http://www.xsol.com/scriptTaskData'

// Returns variable Customer_Name = 'James'
node = dom.createElement (1, 'Customer_Name', '')
root.appendChild (node)
node.text = 'James'

// Returns variable Customer_ID = Null
node = dom.createElement (1, 'Customer_ID', '')
root.appendChild (node)
node.setAttribute ('xsi:nil', 'true')

// Returns variable User = 'UID'
node = dom.createElement (1, 'User', '')
root.appendChild (node)
node.text = '{B48B2229-D7FA-4E8A-96F7-DAFC56199AF0}'

return dom
```




The data returned in each Variable must be of the correct data type for the Variable. For example, returning alpha characters in a Numeric Variable will produce an error.

Changing a Task


Changing a Task involves adding or changing any properties for this item.

To change a Task:


1. Click on the  **Processes** designer button in the **Design for Mapping** window.
2. Click on the required Task.
3. Change any properties in the **Properties Editor**, if required.

Deleting a Task

You can delete a Task in the same way that you can delete any item in *XSOL Mapping* (refer to page 33 for more information). However, deleting a Task in different areas of the application does different things.

-  **Processes** area in the **Design for Mapping** window - Deleting a Task here only deletes the Task assignment to a Stage in a Process. This removes the Task from the Process but not from the model

itself. If you need to remove the Task from the entire model, you need to delete the Task in the  Tasks designer area.

-  area in the **Design for Mapping** window - Deleting a Task here deletes the Task from the entire model. You can only delete the Task if it is not being used in a Process.


4.7.7 Changing the display of Stage/Task nodes

You can change the appearance of Stage and Task nodes in the Design Process Flow window. The following options are available:

- **Show stage icon** - Displays the icon associated with that Stage. All Stages display the same icon.
- **Show stage border** - Displays a visible boundary box around the Stage. Refer to page 29 for more information about changing the size of a boundary box.
- **Show task icon** - Displays the icon associated with that Task. Refer to page 91 for more information about changing Task icons.
- **Show task border** - Displays a visible boundary box around the Task. Refer to page 29 for more information about changing the size of a boundary box.

The chosen settings apply to all Stages and Tasks throughout the model - you cannot have individual settings for a particular Process Flow.

To change the display of Stage/Task nodes:

1. Click on the  icon at the top of the **Design Enterprise** window.
2. Click on the tick boxes described above in the **Properties Editor**. The changes are immediately applied to the model.

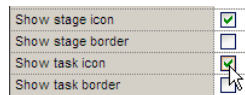


Figure 4.43: Changing the display of Stage/Task nodes

4.7.8 Using Process Summaries



Automation Systems Design only

Process Summaries provide a means of viewing Process instances in an *XSOL Automation* system. All Process Summaries display information in a spreadsheet format, to which you can add columns for more detailed information, and you can create read-only Worksheets that display information about a Process in a form. A Process Summary can optionally drill-down to Stage details, and Stage details can optionally drill-down to Task details.



Process Summary properties

- **Default Label Alignment** - Select the standard label alignment used in any Worksheets.
- **Reference Font Name** - Select the default font that is used in any Worksheets.
- **Reference Font Size in Pts** - Select the default font size that is used in any Worksheets.



Variable/System Value properties:

- **Inherit Label?** - If ticked the item (as a column heading) uses the same caption as the source item. Since long item names can take up a lot of screen real-estate in *XSOL Automation* it is a good idea to shorten item labels.
- **Default Width?** - Enter the default column size, in millimeters, for this item.

To add a Process Summary:

1. Click on the  **Process Summaries** designer button in the **Design for Mapping** window.
2. Click on the Process to which you want to add a Process Summary in the left pane.
3. Click on the  button.
4. Enter the name of the Process Summary and press the **Enter** key to create another item or press the **Tab** key to only create this item.
5. Select any properties for the Process Summary from the **Properties Editor**, if required.
6. You can either add Process Variables or System Values (which equate to columns in *XSOL Automation*) and/or add a read-only Worksheet to the Process Summary.



If you want to add Process Variables and/or System Values to the Process Summary:

- (a) Click on the  **Variables** or  **System Values** tab in the right pane.
- (b) Drag and drop as many items as required to the **Variables in Process Summary entry** folder in the middle pane. The order of the items, from top to bottom, dictates how the columns appear in *XSOL Automation* from left to right. You can re-arrange the order of items by clicking and dragging it to the required position.
- (c) To create a drill-down to the Stage and/or Task level, drag and drop the required Process Variables and/or System Values to the **Variables in Stage Summary entry** and/or **Variables in Task Summary entry**.



You need at least one Variable or System Value at the Process-level to activate the Stage drill-down in *XSOL Automation*, and at least one Variable or System Value at the Stage-level to activate the Task drill-down.

If you want to add a Worksheet to the Process Summary:

- (a) Click on the  button in the middle pane and enter the title of the Worksheet.
- (b) Click on the  button.
- (c) You need to create new widgets and/or drag and drop variables onto the worksheet to create display items. Refer to page 107 and page 110 for more information.

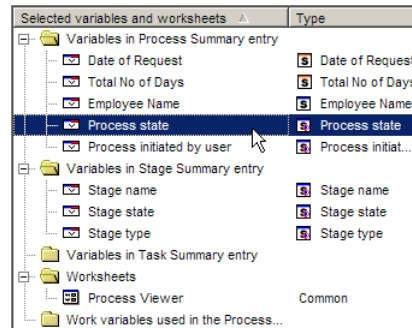



Figure 4.44: Adding a Process Summary with a Stage drill-down

7. If you need to set the sort order of any of the Variables as they appear in the run-time system, click on the  button.
8. Click on the **Sort Direction** column next to the Variable that you want to order. Select one of the following options:
 - **Ascending** - The column is sorted in ascending sequence - alphabetical order, 1-10.
 - **Descending** - The column is sorted in descending sequence - reverse-alphabetical order, 10-1.
 - **None** - The column is not sorted in any specific order.

Repeat for as many Variables as required, and click on the **OK** button when you have finished.

9. Add or change any properties in the **Properties Editor**, if required.
10. You now need to customise the default Process Summary Filter that was automatically created with the Process Summary - refer to page 77 for more information.


Changing a Process Summary



Automation Systems Design only

Changing a Process Summary involves altering the Variables and/or the Worksheets displayed in the Process Summary.

To change a Process Summary:

1. Click on the  **Process Summaries** designer button in the **Design for Mapping** window.
2. Click on the required Process Summary.
3. Add or edit the Process Variables and/or Worksheets for the Process Summary.

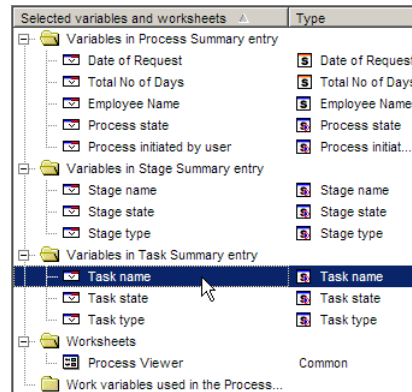



Figure 4.45: Changing a Process Summary

If you need to set the sort order of any of the Variables as they appear in the run-time system, click on the  button.

Using Process Summary Filters



Automation Systems Design only




Process Summary Filters allow you to specify what information a particular user can see in a Process Summary and whether they can customise that information. A Process Summary can have any number of filters, each with its own Roles assigned and customised properties.

Process Summary Filter properties

- **Default Execution Level** - Identifies whether the Roles assigned to the Filter can (**No one except assigned Roles**) or cannot (**Everyone except assigned Roles**) initiate the Process Summary Filter.
- **Allow runtime customisation of** - Tick any of the tick boxes if you want users to be able to change filter information in *XSOL Automation*.
- **Process state(s)** - Tick any of tick boxes to select which Process states you allow users to view.
- **Stage state(s)** - Tick any of tick boxes to select which Stage states you allow users to view.
- **Event type(s)** - Tick any of tick boxes to select what event information (such as Task types or default values) you allow users to view. These properties are only available if the Filter has one or more Variables in the **Variables in Task Summary entry** folder.
- **Process started in period** - Select one of the following options:
 - Time Period - Includes only Processes that were started in the entered period. Enter the required number in the **Time Period** field followed by **d** (day), **w** (week), **m** (month), or **y** (year).
 - Financial Period - Includes only Processes that were started in the selected period. These periods are all time periods relative to the Trading Entity financial year (refer to page 39 for more information). Select one of the options in the **Time Period** drop-down list.
- **Process finished in period** - Select one of the following options:
 - Time Period - Includes only Processes that were finished in the entered period. Enter the required number in the **Time Period** field followed by **d** (day), **w** (week), **m** (month), or **y** (year).

- Financial Period - Includes only Processes that were finished in the selected period. These periods are all time periods relative to the Trading Entity financial year (refer to page 39 for more information). Select one of the options in the **Time Period** drop-down list.
- **Stage scheduled in period** - Select the required Stage from the **Select stage** drop-down list and select one of the following options:
 - Time Period - Includes only Stages that were scheduled in the entered period. Enter the required number in the **Time Period** field followed by **d** (day), **w** (week), **m** (month), or **y** (year).
 - Financial Period - Includes only Stages that were scheduled in the selected period. These periods are all time periods relative to the Trading Entity financial year (refer to page 39 for more information). Select one of the options in the **Time Period** drop-down list.
- **Stage executed in period** - Select the required Stage from the **Select stage** drop-down list and select one of the following options:
 - Time Period - Includes only Stages that were executed in the entered period. Enter the required number in the **Time Period** field followed by **d** (day), **w** (week), **m** (month), or **y** (year).
 - Financial Period - Includes only Stages that were executed in the selected period. These periods are all time periods relative to the Trading Entity financial year (refer to page 39 for more information). Select one of the options in the **Time Period** drop-down list.
- **Only processes initiated by** - Select the type of user who initiated the Process from the drop-down list.
- **Only selected stage and/or initiated by** - Select the Stage that was initiated from the **Select stage** drop-down list and select the type of user who initiated the Stage from the **Initiator** drop-down list.

To add a Process Summary Filter:

1. Click on the required Process Summary in the **Design for Mapping** window.
2. Click on the  button.
3. Enter the name of the Process Summary Filter and press the **Enter** key to create another item or press the **Tab** key to only create this item.
4. Select any properties for the Process Summary Filter from the **Properties Editor**, if required.
5. Add one or more Roles to the Process Summary Filter, either by clicking on the  button or dragging and dropping existing Roles from the  button in the right pane.

Caption	Customer Complaints
Default Execution Level	No one except assigned Roles
Allow runtime customisation of	Process state(s), Stage state(s)
Process state(s)	<input checked="" type="checkbox"/>
Stage state(s)	<input checked="" type="checkbox"/>
Process started in period	<input type="checkbox"/>
Process finished in period	<input type="checkbox"/>
Stage scheduled in period	<input type="checkbox"/>
Stage processed in period	<input type="checkbox"/>
Process initiated by	<input type="checkbox"/>
Stage initiated by	<input type="checkbox"/>

Figure 4.46: Setting the properties of a Process Summary Filter

Changing a Process Summary Filter



Automation Systems Design only

Changing a Process Summary filter involves adding or changing any properties as well as adding or removing Roles.

To change a Process Summary Filter:

1. Click on the required Process Summary in the **Design for Mapping** window.
2. Click on the required item in the list.
3. Change any properties in the **Properties Editor**, if required.

4.8 Designing Process Flow

The sequence in which Stages are performed within a Process, or Tasks are performed within a Stage. Process Flow permits consecutive, concurrent, and conditional flow to be defined; once a Process has been initiated the XSOL Process Manager controls the what-happens-next scenario, using the Roles assigned to the Stages to achieve this.



There are several types of Process Flow:

1. **Stage Process Flow** - Process Flow within a Process. Stage Process Flow can be scheduled.
2. **Task Process Flow** - Process Flow within a Stage. Stage cannot be scheduled.

Process Flow uses several types of conditions and other Process Flow nodes, including If, All, and Any conditional links, to help you exactly specify Process Flow. Refer to page 80 for more information about Process Flow nodes.

You can use Swim Lanes in the Process Flow diagram, if required (refer to page 89 for more information).

To design a Process Flow diagram:

1. Click on the  **Processes** designer button in the **Design for Mapping** window.
2. Click on the required Process.
3. Click on the  button.
4. You are now in the **Design Process Flow** window. Refer to page 80 for more information about designing Business Processes.

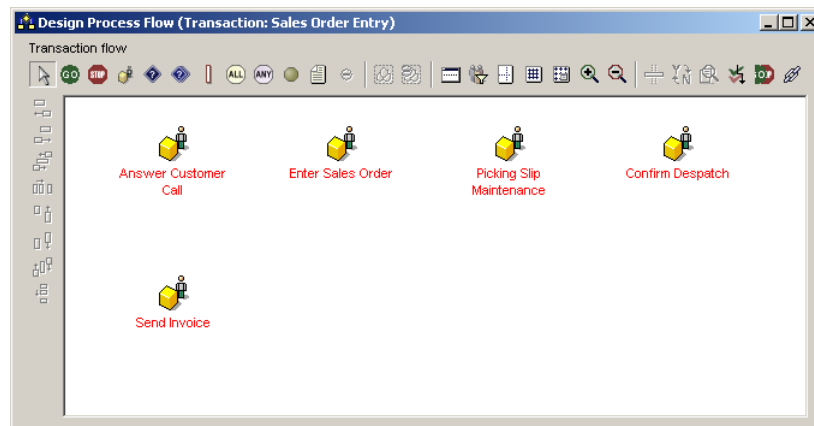














Figure 4.47: Designing Process Flow

4.8.1 Adding Process Flow nodes

Process Flow nodes allow you to control the path of a Business Process. Many of these nodes are conditions that allow you to state that different actions occur based on different circumstances. For example, in an Order Entry Process, all orders over \$10,000 can be directed to the credit controller for approval, while all orders under \$10,000 are automatically approved and proceed directly to the Pick Order stage.

There are several different types of nodes available for Process Flow:

-  **Select** - Allows you to select items.

-  **Start** - No input link, one output link. There can only be one Start node per Process Flow.
-  **Stop** - One input link, no output link. You can have multiple Stop nodes per Process Flow.
-  **Stage** - Adds a new Stage to the Process Flow. This will automatically update the Stages in the rest of the application. Only available in Stage Process Flow.
-  **Task** - Adds a new Task to the Process Flow. This will automatically update the Tasks in the rest of the application. Only available in Task Process Flow.
-  **Condition** - One input link, one output link, with one Yes/No attached condition. A Condition node always has two output paths - one for Yes and one for No. Conditions always have an attached Expression (refer to page 138 for more information).
-  **Decision** - One input link, multiple output links, with multiple Yes/No attached conditions. A Decision is a convenient way of putting several Condition nodes together in one place in your Process Flow. All outputs occur concurrently. Decisions always have an attached Expression (refer to page 138 for more information).
-  **Fork** - One input link, multiple output links, with no attached conditions. All outputs occur concurrently.
-  **All** - Multiple input links, one output link, with no attached conditions. All inputs must occur before the output will be triggered.
-  **Any** - Multiple input links, one output link, with no attached conditions. The first input to arrive will trigger the output.
-  **Joint** - Multiple input links, one output link. Links two or more Process Flow nodes together.
-  **Annotation** - Adds an Annotation to the Process Flow. Annotations allow you to enter notes anywhere on the Process Flow diagram, and can be scaled to the required size.
-  **Swim Lane** - Adds a Swim Lane to the Process Flow diagram - refer to page 89 for more information.
-  **Delete** - Deletes the selected node or link. You can delete multiple nodes or links at a time, if required.

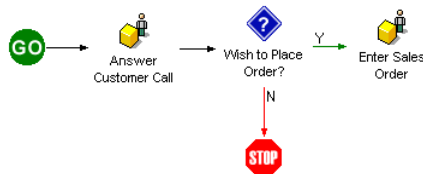
Example:

Figure 4.48: Process Flow diagram

Adding a Start/Stop node

Start and Stop nodes are used to begin and end a Process Flow diagram and are a necessary part of designing a valid Process Flow diagram. You need one Start node and at least one Stop node to create a valid Process or Task Process Flow diagram.



Start node rules:

- One output link
- No input links
- One must exist in a valid Process Flow diagram

Stop node rules:

- One input link
- No output links
- At least one must exist in a valid Process Flow diagram

To add a Start or Stop node:

1. Open the **Design Process Flow** window for the required Process.
2. Click on the  or  button.
3. Click on the space in the work area where you want to insert the node.
4. Repeat the procedure for multiple nodes, if required.
5. Link the node to other nodes. Refer to page 94 for more information about linking new nodes together.


Adding a Stage/Task node

Stage and Task nodes are the Process Flow representation of Stages and Tasks that you have created in the **Design for Mapping** window (refer to page 54 and page 60 for more information). You can create new Stages and Tasks while designing your Process Flow diagram, if required.

Rules:

- One input link
- One output link
- At least one Stage must exist in a valid Stage Process Flow diagram and at least one Task must exist in a valid Task Process Flow diagram

To add a Stage or Task node:

1. Open the **Design Process Flow** window for the required Process.
2. Click on the  button.
3. Click on the space in the work area where you want to insert the node.
4. Fill in the Stage or Task details in the pop-up window. The minimum amount of information that you need to enter is a meaningful name in the **Caption** field.
5. Repeat the procedure for multiple nodes, if required.
6. Link the node to other nodes. Refer to page 94 for more information about linking new nodes together.

Adding a Joint node

Joint nodes are used to merge one flow with multiple possible routes into one Stage or Task. Joints should only be used as a result of a Condition. A Joint node is necessary because Stages and Tasks can only have one input, not multiple inputs.

The following diagram shows two separate flows merging into one Stage using a Joint node.

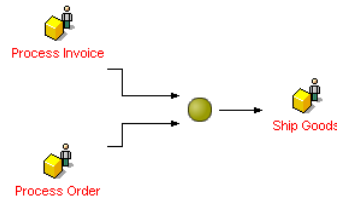


Figure 4.49: Merging two separate flows

The following diagram shows a repeat loop using a Condition node and a Joint node.

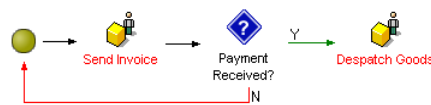



Figure 4.50: Repeat loop

Rules:

- Multiple input links
- One output link

To add a Joint node:

1. Open the **Design Process Flow** window for the required Process.
2. Click on the  button.
3. Click on the space in the work area where you want to insert the node.
4. Repeat the procedure for multiple nodes, if required.
5. Link the node to other nodes. Refer to page 94 for more information about linking new nodes together.

Adding an All/Any node

All and Any nodes are almost identical. Both accept multiple input links and have one output link. The only difference is that an All node needs all input links to occur before triggering the next action, whereas an Any node only needs one input link to occur before triggering the next action.

The following diagram shows that to proceed to the "Despatch Goods and Invoice" Stage, both the "Print Invoice" and "Pick Goods" Stages must have been completed.

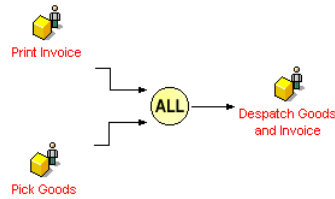


Figure 4.51: All Stages must be completed for an All node

The following diagram shows that to proceed to the "Despatch Goods or Invoice" Stage, only one of the "Print Invoice" and "Pick Goods" Stages must have been completed.

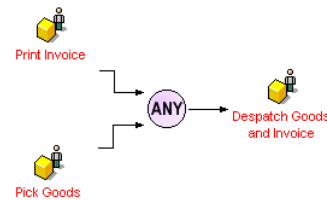


Figure 4.52: Only one Stage must be completed for an Any node



All node rules:

- Multiple input links
- One output link
- All input links must occur to trigger the output link

Any node rules:

- Multiple input links
- One output link
- One input link must occur to trigger the output link

To add an All or Any node:

1. Open the **Design Process Flow** window for the required Process.
2. Click on the  or  button.
3. Click on the space in the work area where you want to insert the node.
4. Repeat the procedure for multiple nodes, if required.
5. Link the node to other nodes. Refer to page 94 for more information about linking new nodes together.

Adding a Condition node

A Condition node has two outputs based on a condition. For example, the following diagram shows that there is one path if payment has occurred, and one path if payment has not occurred.

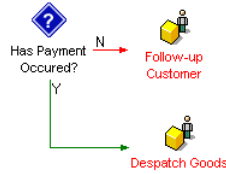



Figure 4.53: A Condition node


Rules:

- One input links
- Two output link
- Which output link is triggered depends upon the Condition.

To add a Condition node:

1. Open the **Design Process Flow** window for the required Process.
2. Click on the  button.
3. Click on the space in the work area where you want to insert the node.
4. Fill in the Condition node details in the pop-up window. The minimum amount of information that you need to enter is a meaningful name in the **Caption** field.
5. Repeat the procedure for multiple nodes, if required.
6. Link the node to other nodes. Refer to page 94 for more information about linking new nodes together.

Once you have linked the Condition node to two other nodes you can set the text for the Y and N links. Double-click on the Condition node to set this text in the **Caption for False** and **Caption for True** fields. You can set several options for caption text and which link is drawn first - refer to page 34 for more information.

You can swap the captions between the links by clicking on the  button.

Adding a Decision node

A Decision node allows you to have two or more outcomes based on a number of conditions. For example, the following diagram shows three possible options based on the value of an order.

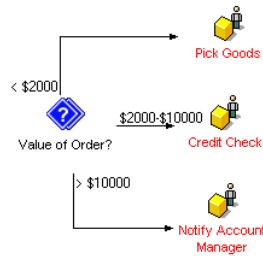




Figure 4.54: A Decision node

Rules:





- One input links
- Two or more output link
- Which output link is triggered depends upon various condition


To add a Decision node:

1. Open the **Design Process Flow** window for the required Process.
2. Click on the  button.
3. Click on the space in the work area where you want to insert the node.
4. Fill in the Decision node details in the pop-up window. The minimum amount of information that you need to enter is a meaningful name in the **Caption** field.
5. Link the Decision node to other nodes. Refer to page 94 for more information about linking new nodes together. You can link to as many nodes as required from a Decision node.
6. If this model is for an *XSOL Automation* system you now need to set the Expressions and outcomes for the Decision node. Open the Expansion Panel by clicking on the  button.



If you have added the Decision node to Task Process Flow you need to move the Expansion Panel to the top of the window by clicking on the  button, and then clicking on the Decision node.

7. Click on the  button to create a new condition and name it appropriately.
8. Double-click on the Expression column next to the first condition to create the Expression.
9. Enter the required Expression - refer to page 138 for more information.
10. Click on the  button to create a new decision and name it appropriately.
11. Choose one of the following options for the decision column next to the condition to indicate how *XSOL Automation* Processes this condition:
 -  - Processes the result of this condition if it is true.
 -  - Processes the result of this condition if it is false.
 - ☐ - Ignores the result of this condition.

12. Choose one of the following options for the decision column next to the outcome to indicate which links *XSOL Automation* follows based on the results of the proceeding conditions:
- ☒ - Follows this link to the next node.
 - ☐ - Does not follow this link to the next node.
13. Click on the  button to create a new condition, if required, and repeat the above steps to set the Expression and outcomes for the condition.
















Name	Expression	 Pricing	 Oversupply	 Undersupply	 Damaged	 Incorrect
Conditions						
 Pricing issue?	return Is_it_a_pricing_issue	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 Oversupply?	return Was_it_an_oversupply	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 Undersupply?	return Was_it_an_undersupply	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 Damaged?	return Was_it_damaged_goods	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 Incorrect?	return Was_it_incorrect_product	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Outcomes						
 1		<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 2		<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 3		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
 4		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
 5		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 4.55: Outcomes for a Decision Table

14. You can now set the text for the links, if required. Click on the required outcome and enter the title into the **Caption** field in the **Properties Editor**. Repeat for each condition as required.

Adding a Fork node

A Fork node allows you to start parallel flow. A Fork node can initiate two or more flows that run at the same time. For example, the following diagram demonstrates an "Input Order" stage that initiates "Print Invoice" and "Pick Goods" Stages at the same time.

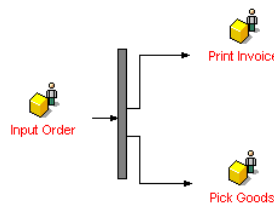




Figure 4.56: A Fork node

Rules:

- One input links
- Two or more output link
- All output links are initiated at the same time

To add a Fork node:


1. Open the **Design Process Flow** window for the required Process.


2. Click on the  button.
3. Click on the space in the work area where you want to insert the node.
4. Alter the appearance of the node, if required.
 - To flip a Fork from horizontal to vertical, or vice versa, click on the Fork node and click on the  button.
 - To shorten or lengthen a Fork, Shift+click on one end of a Fork and drag it to the required length.
5. Repeat the procedure for multiple nodes, if required.
6. Link the node to other nodes. Refer to page 94 for more information about linking new nodes together.

Adding an Annotation node

Annotations allow you to enter notes anywhere on the Process Flow diagram, and can be scaled to the required size. You can change the background colour of this or all annotation nodes, if required - refer to page 169 for more information.

To add a Annotation node:

1. Open the **Design Process Flow** window for the required Process.
2. Click on the  button.
3. Click on the space in the work area where you want to insert the node.
4. Enter the text for the Annotation node in the **Text** property.

If you need to enter more than one line of text, click on the  button and enter the text in the editor. Click on the **OK** button to save the Annotation node.
5. Re-size and move the node to a better position, if required. To re-size a node hold down the **Shift** key and click and drag on the outer sides or corners.
6. If you need to change the background colour of the annotation node, select the required colour from the **Background color** drop-down list property in the **Properties Editor**.
7. Repeat the procedure for multiple nodes, if required.

Adding a Group node


Groups are a convenient way of representing large amounts of flow within one node without taking up large amounts of space.

To add a Group node:

1. Select the nodes that you want to include in the Group. All nodes need to be in the same piece of Process Flow; they have to be linked together.



Refer to page 96 for more information about selecting multiple nodes.

2. Click on the  button.

3. Double-click on the new Group node to change any properties of the Group, such as the title of the node or whether it will print in any generated documentation.

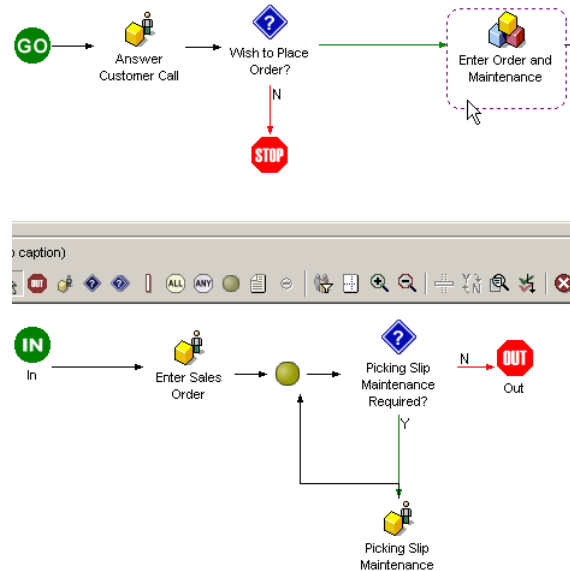


Figure 4.57: Adding a Group node

4.8.2 Using Swim Lanes

Swim Lanes add an additional visual layer to Process Flow diagrams. Swim Lanes add vertical or horizontal columns to a diagram, and are often used to delineate ownership of all Stages within the Swim Lanes.



Swim Lanes can only be added to Stage Process Flow diagrams (not Task Process Flow diagrams).

To add a Swim Lane:

1. Right-click on the in the required Stage Process Flow diagram and click on the **Insert >New Swim Lane** menu option..
2. Re-name the caption for the Swim Lane.
3. Alter the appearance of the Swim Lane, if required.
 - To change the orientation of Swim Lanes, right-click on the Process Flow diagram and click on the **Swim Lanes Orientation >Horizontal** or **Swim Lanes Orientation >Vertical** menu item.
 - To change the width of a Swim Lane, hold down the **Shift** key and click and drag the required border to its new position.
 - To move a Swim Lane, click on the Swim Lane's caption and drag it to its new location.
 - To change the background colour of a Swim Lane, click on the required colour from the **Colour** property in the **Properties Editor**.

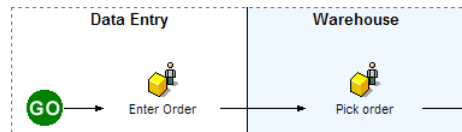



Figure 4.58: Using Swim Lanes

4.8.3 Opening Task Process Flow

You need to open the Expansion Panel in order to view Task Process Flow.

To open Task Process Flow:

1. Click on the  button in the **Design Process Flow** window.
2. Click on the required Stage.

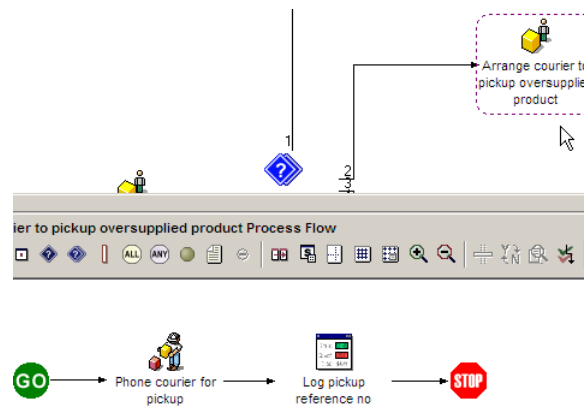




Figure 4.59: Opening Task Process Flow



You can move the Expansion Panel to the top of window by clicking on the  button, and move it back down by clicking on the  button. Usually you only need to move the Expansion Panel to the top of the window when you are using Decision nodes in Task Process Flow - refer to page 85 for more information.



4.8.4 Automatically creating Process Flow diagrams

There are two different functions that allow you to quickly create linked and valid Process or Task Process Flow. You can do any of the following:

- Generate Start and Stop nodes automatically
- Generate links between nodes automatically (in the order of creation)

Both of these functions only work the first time a Process Flow diagram is opened. Both functions are best used when you create a list of Stages in a Process, or Tasks in a Stage, in the correct order, then open up the Process Flow diagram and create the Start and Stop nodes and the links between the nodes.

To automatically create Process Flow diagrams:

1. Add several Stages to a new Process. Refer to page 54 for more information about adding Stages.
2. Open the **Design Process Flow** window for the Process.
3. Click on as many of the following buttons as required:
 -  - Adds Start and Stop icons to the beginning and the end of the Process Flow.
 -  - Automatically links all Stages or Tasks together sequentially (in the order of creation).

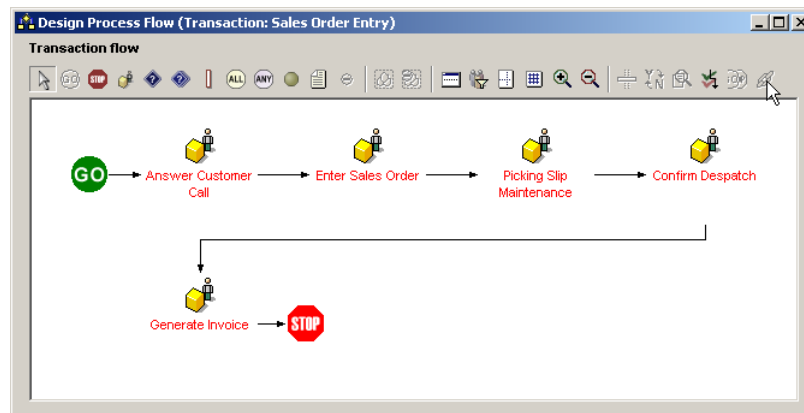


Figure 4.60: Automatically creating a Process Flow diagram

4.8.5 Changing a Process Flow icon

You can change the displayed icon for any Process Flow node in a diagram. This allows you to represent visually what type of action or logic is being performed. You can also reset all icons that have been manually changed to the application defaults.



Refer to page 92 if you want to change the appearance of all nodes of a particular type, rather than one individual node.

To change a Process Flow icon:

1. Click on the node that you want to change in the Process Flow diagram.
2. Click on the **Icon** field in the **Properties Editor**.
3. Click on the ... button.
4. Click on the icon that you want to use for this node.

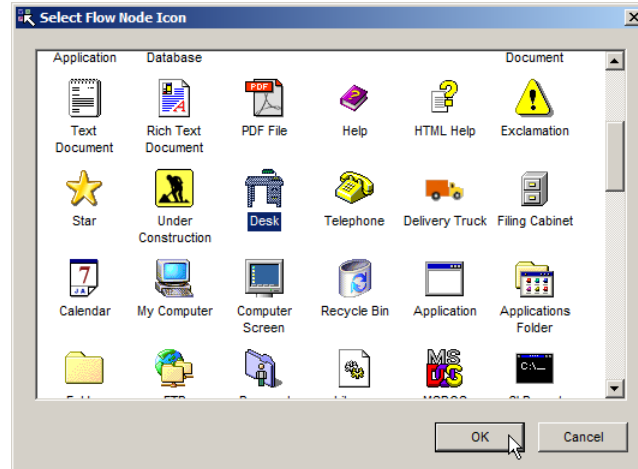



Figure 4.61: Changing a Process Flow icon

5. Click on the **OK** button.
6. To reset all icons that have been manually changed to the application defaults:
 - (a) Click on the  icon at the top of the **Design Enterprise** window.
 - (b) Click on the **Reset individual node icons** property and click on the ... button.
 - (c) Click on the **Yes** button. This will change all icons that have been manually changed to the defaults that are displayed under the **Default flow control icons** and **Default task icons** headings.


4.8.6 Changing the default icon for a Process Flow node

You can change the default icon used by any Process Flow node. This allows you to visually standardize your Process Flow diagrams. You can save these changes as the new default icons for the application, restore these defaults if unintentionally changed, or restore the XSOL default icons shipped with the *XSOL Mapping* software. Refer to page 178 for another means of setting default icons for the application.



This does not change existing nodes, only new nodes that are created after you made the change to the default settings. Refer to page 91 if you only want to change the appearance of one individual node, rather than all nodes of a particular type.

To change the default icon for a Process Flow node:

1. Click on the  icon at the top of the **Design Enterprise** window.
2. Click on the field that you want to change in the **Properties Editor**. All Process Flow icon fields are displayed underneath the **Default flow control icons** and **Default task icons** headings.
3. Click on the ... button.
4. Click on the icon that you want to use for this node.

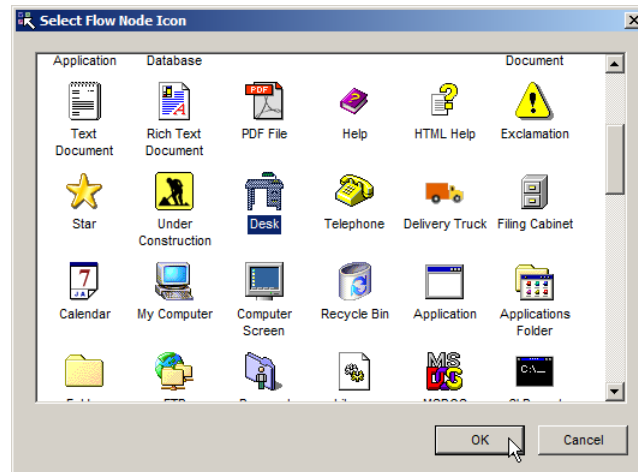




Figure 4.62: Changing a Process Flow icon


5. Click on the **OK** button.



All manually changed nodes will not be effected by this change - refer to page 91 for more information.








6. If you need to set the current default icons as the permanent default icons, click on the  button in the **Save or Reset application defaults** property, select the **Save as application defaults** radio button, and click on the **OK** button.




If you need to restore the default icons, click on the  button in the **Save or Reset application defaults** property, select the **Reset to application defaults** radio button, and click on the **OK** button.

If you need to restore the default XSOL icons shipped with the *XSOL Mapping* software, click on the  button in the **Save or Reset application defaults** property, select the **Reset to XSOL defaults** radio button, and click on the **OK** button.




4.8.7 Aligning Process Flow nodes

The **Design Process Flow** window provides you with several functions that help you align Process Flow nodes on the screen in a consistent manner. The following functions are available:

-  - Displays a grid. The grid allows you to align items properly by using the grid as a visual frame of reference.
-  - Snap to grid. Automatically aligns the top left corner of the selected node to the nearest grid markers.
-  - Move all selected nodes in line with the far left node.
-  - Move all selected nodes in line with the far right node.
-  - Move all selected nodes in line with the horizontal central point between all nodes.
-  - Move all selected nodes so that they are equally spaced horizontally.
-  - Move all selected nodes in line with the top node.

-  - Move all selected nodes in line with the bottom node.
-  - Move all selected nodes in line with the vertical central point between all nodes.
-  - Move all selected nodes so that they are equally spaced vertically.

To align Process Flow nodes:

1. To turn the grid on or off, click on the  button.
2. To enable or disable the Snap To Grid function, click on the  button.
To set the size of the grid, click on the small down arrow next to the  button.
3. To use any of the alignment buttons, select more than one node (click and drag a box around the required nodes, or Ctrl+click on the required nodes), and click on the required button.

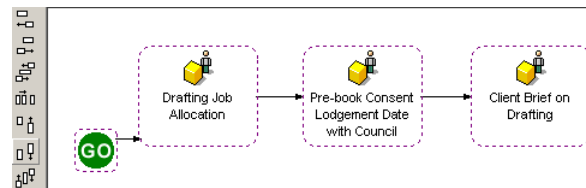


Figure 4.63: Aligning the bottom of selected nodes

4.8.8 Linking Process Flow nodes

All nodes in a Process Flow, including Stages, Tasks, Ifs, Ors, etc., must be linked together to be included in the Process Flow. Linking simply involves dragging a connector line between two Process Flow nodes.

To link Process Flow nodes:

1. Move the mouse cursor over the node that you want to link from in the **Design for Mapping** window. The node is now highlighted in yellow.
2. Click on the yellow bar on the side of the node that you want the link to come from and hold down the mouse button.
3. Drag a link to the side of the node that you want to link to. The node will be highlighted with yellow bars.

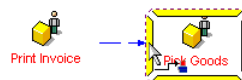



Figure 4.64: Linking Process Flow nodes

4. Let go of the mouse button.

4.8.9 Changing Process Flow node properties

You can change the properties for a Process Flow node in the **Design for Mapping** window. This is exactly the same as changing the Properties for that node in its original window - it is simply a convenient way of altering an item's properties.

To change a Process Flow node's properties:

1. Click on the node that you want to change in the **Design for Mapping** window.
2. Click on the  button.



You can also double-click on the node to immediately open the **Editing** window.

3. Change the properties for that item. Refer to other parts of this manual if you need more information about the properties of various nodes.


Property	Value
Caption	Raise Credit
Stage Summary for To-do List	return toString ("Client: ", Cus...
Schedule Expression	return Date_of_Complaint - ...
Reference	
To-do Values Action	After process values
Runtime Hint	
Version Note	
Designer comments	
Help URL	
External ID	Raise_Credit

Figure 4.65: Changing a Process Flow node's properties

4.8.10 Validating Process Flow links

Validating links is an important part of making sure your model is logically correct. For example, a Start node must have one link leading from it and no links leading to it. You can validate Process Flow or Task Process Flow at any time to check that you have created or changed your Process Flow correctly.

To validate Process Flow links:

1. Click on the  button in the required Process or Task Process Flow window, or press **Alt+V**.
2. A dialog box with an appropriate message tells you if there are any invalid links.

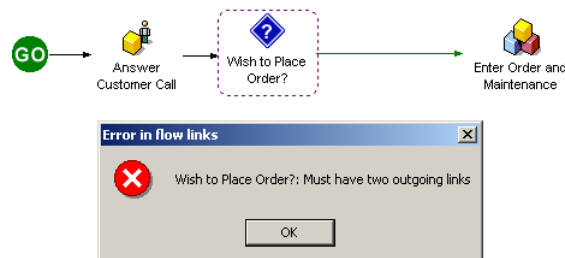


Figure 4.66: Validating Stage Process Flow links

4.8.11 Selecting Process Flow nodes

XSOL Mapping allows you to select Process Flow nodes individually or as a group. This enables you to move part of your Process Flow diagram around as a group, which is helpful when you need to insert a new node in the middle of your diagram or to produce a more balanced diagram.

To select a Process Flow node(s):

1. Click on the node with your mouse. If you need to select multiple nodes, click and drag your mouse around the nodes or click on each node while holding down the Ctrl key.
2. Click on the selected node or group and drag it to another part of the work area.

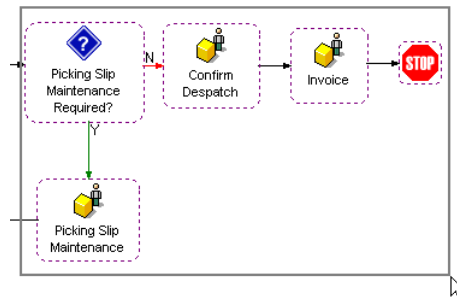
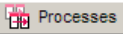


Figure 4.67: Selecting multiple Process Flow nodes

4.8.12 Viewing Process Flow

There are several different ways you can change your view of the Process Flow window:

- **Role information** - Displays the Roles assigned to a Stage. This is a summarised version of the information available in the  button in the **Design for Mapping** window. You can view this information either as a tooltip or directly on the diagram itself (which will be included in the generated documentation).
- **Resize to fit** - Resizes the selected node(s) to best fit the text in the node. This is particularly useful when role information is displayed for Stages.
- **Expansion panel** - Displays the Task Process Flow diagram within a selected node in a Stage Process Flow diagram. You can design Task Process Flow in exactly the same way as Stage Process Flow, except that you are using Tasks instead of Stages.
- **Zooming** - Zoom out if you have a large Process Flow diagram and need to see how it flows together, or zoom in if you cannot see enough detail.






While the Process Flow diagram is zoomed you can move nodes around and change links. However, you cannot resize nodes, move link labels, or shift existing links to other nodes.

- **Grid lines** - Grid lines give you a visual frame of reference that enable you to line up items in a straight line.
- **Page Boundaries** - Page Boundaries displays where a Process Flow diagram fits on a page, and how many pages it requires in the printed documentation (if more than one). Page Boundaries alter depending on what page size you have chosen for a diagram and whether the diagram is orientated in portrait or landscape mode.

To change the Process Flow window view:

1. Follow one or more of the following instructions:

- To **view** the Role information for a Stage as a tooltip, place your cursor over the Stage.
- To **view** the Role information for a Stage on the diagram itself, click on the  button or select the **Show role assignments** tick box in the **Properties Editor** for the diagram. The default setting for this tick box can be set at the model and application levels - refer to page 34 and/or page 170 for more information.
- To **resize** one or more nodes, select the required nodes (using click, **Shift+click**, or **Ctrl+click**) and click on the  button. You may need to alter the position of the nodes in the diagram after resizing.
- To **display or hide** the Expansion Panel, click on the  button.

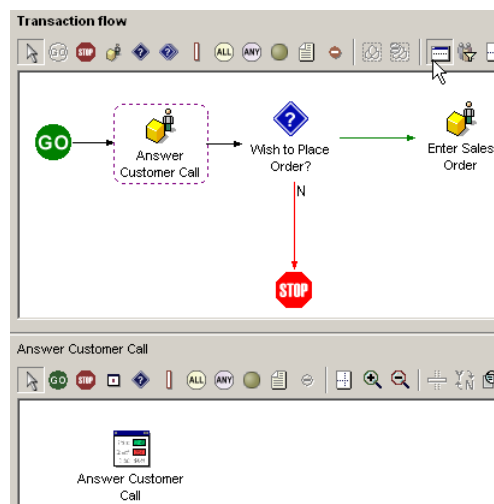




Figure 4.68: Closing the Expansion Panel


- To **zoom in or out** of the diagram, click on the required item in the zoom drop-down list. The **Cancel zoom** option returns you to the 100% view.
- To **display or hide** grid lines, click on the  button.
- To **display** the page boundaries for this diagram, click on the  button.

4.8.13 Viewing available items in Task Process Flow

XSOL Mapping enables you to view various items that are available to the Task initiator in Task Process Flow. The following items are available for display:

- Processes
- Reference Variables

To view the available items in Task Process Flow:

1. Open the **Design Process Flow** window and click on the Stage that you want to view.
2. Open the Expansion Panel by clicking on the  button, if it is not open already.

3. Click on one of the following buttons:

-  - Opens a pop-up window with a list of the Processes that are available to the initiator of this Process.

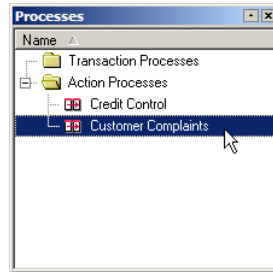



Figure 4.69: Creating a new Initiate process Task

If you drag and drop one of the displayed items onto the Stage, a new **Initiate process** Task will be created linked to the selected Process.

-  - Opens a pop-up window with a list of the Reference Variables that are available to the initiator of this Process.

If you drag and drop one of the displayed items onto the Stage, a new **Table Update** Task will be created linked to the Reference Variable.


4.8.14 Filtering Process Flow by Role

You can filter a Process Flow diagram by Role. This function holds the titles of Stages that are used by the Role or Roles selected, allowing you to quickly see what Stages are performed by what Roles.



Filtering is not permanent - the filtering only lasts as long as the **Design Process Flow** window is open.

To filter Process Flow by Role:

1. Open the **Design Process Flow** window for the Process Flow diagram that you want to view.
2. Click on the  button.
3. Click on the tick boxes next to the Roles that you want to view.
4. Click on the **Filter** button. The Stage titles that are performed by that Role are now bold.

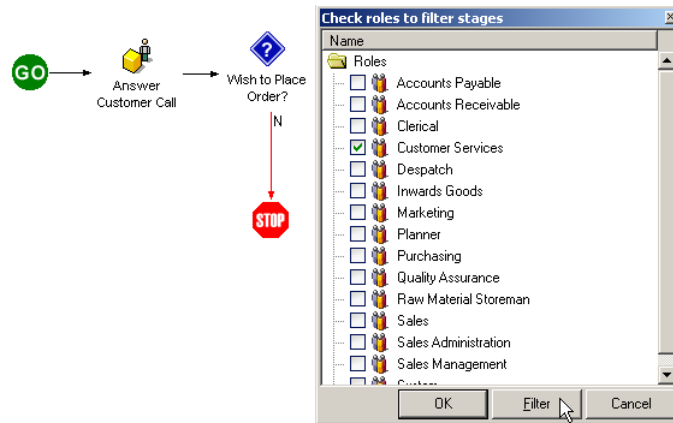



Figure 4.70: Filtering a Process Flow diagram by Role

- Click on the **OK** button when you are have finished with the window.

4.8.15 Printing a Process Flow diagram

You can print the Process Flow diagram you are currently viewing. The print option lets you preview the diagram before you print it, choose the printer you want to use, save the diagram as a graphic, and set the magnification level for both the preview and the printed diagram.

To print a Process Flow diagram:

- Open the Process Flow diagram that you want to print.
- Click on the  button.
- Click on any of the options, if required.
 - Page setup...** - Allows you to set the paper size, source, orientation, and margins for the printed diagram.
 - Save as...** - Allows you to save the diagram as a graphic (PNG, EMF, BMP, and JPG file types are supported).
 - Page** - Use the drop-down list or navigation buttons to move between the diagram pages.
 - Print scale** - Select the magnification option for the printed diagram.
 - Preview scale** - Select the magnification option for the displayed diagram.

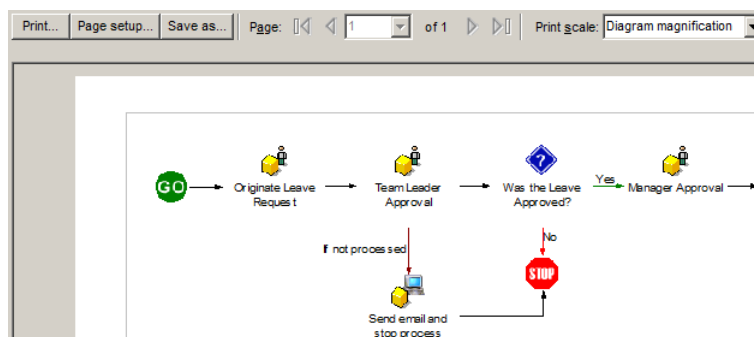


Figure 4.71: Printing a Process Flow diagram

4. Click on the **Print...** button to open the **Print** dialog.
5. Select the required printer and specify any other options that are required.
6. Click on the **OK** button.


4.8.16 Printing options for documentation generation

You can set a number of different printing options for both Stage and Task Process Flow diagrams which alter how a diagram will appear in the generated documentation. You can set the following properties:

- **Print In Documentation** - Indicates whether this diagram is included in the generated documentation or not.
- **Diagram Magnification** - Sets the scale of the diagram in percentile.
- **Paper Size** - Sets the type of paper used for printing the diagrams.



Only A4 paper is supported.


- **Rotation for flow diagrams** - Sets the logical "up" position for the diagram. If you are sitting on a chair, the **Portrait** option sets the "up" position towards the ceiling, while **Landscape** sets the "up" position to your left. These settings do not have anything to do with paper orientation - that is controlled by the documentation option chosen, such as **Microsoft Word (portrait)**.
- **Review Stages** - Opens a separate dialog box that lists all the Stages Process Flow diagrams that will be printed for this Process when you generate documentation. This is a separate button -  - and not a property in the **Properties Editor**.

To set printing options for Process Flow diagrams:

1. Click on white space anywhere on a Process Flow diagram.
2. Alter the options detailed above in the **Properties Editor**.

Print in documentation	<input checked="" type="checkbox"/>
Diagram Magnification	100%
Paper Size	A4
Rotation for flow diagram	Portrait
Show role assignments	<input type="checkbox"/>


Figure 4.72: Setting printing options for Process Flow diagrams

3. If you need to review the Task Process Flow that you want to include in the generated documentation, click on the  button. Select the tick boxes next to the Task Process Flow that you want to include in the documentation.

4.8.17 Using reference captions

A reference caption is a user-defined labeling system that is displayed on Processes, Stages, and Tasks in Process Flow diagrams. Reference captions are added to the Process Flow diagrams automatically. However, the labeling system that you use (numerical, alphabetical, etc.) is your decision and must be entered manually.

To set up and use reference captions:

1. Click on the  icon at the top of the **Design Enterprise** window.
2. Open the **Properties Editor**, if it is not open already.
3. Click on the required option from the **Reference Caption Position** drop-down list. The **None** option disables reference captions.
4. Enter the separator between each reference caption in the **Reference Caption Separator** field.

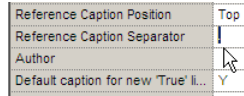


Figure 4.73: Using reference captions

5. For each Process, Stage, and Task that you want to add a reference caption to:
 - (a) Click on the required item in the **Design for Mapping** window.
 - (b) Enter the reference caption for the item in the **Reference** field in the **Properties Editor**. The reference caption is now visible in the Process Flow diagram and is included as part of the generated documentation.

4.9 Designing Roles

The class of user to whom a Stage is assigned who is normally associated with a functional business area; for example, either broad - Accounting - or narrow - Credit Controller. Roles can be assigned to the Enterprise or Trading Entities.


Roles are items that can be reused within a model. Each time you create a Role within a Process you are creating a parent Role within the Enterprise or Trading Entity.

For an expanded definition please refer to page 17 for more information.

Role properties:

- **Access level** - Determines what access this Role has to this Stage. This overrides the default access level for the escalation - refer to page 56 for more information.

To add a Role:

1. Click on the  button in the **Design Enterprise** or the **Design for Mapping** window.
2. Enter the name of the Role and press the **Enter** key to create another item or press the **Tab** key to only create this item.
3. Add or change any properties in the **Properties Editor**, if required.
4. You may want to assign this new Role to one or more Stages (refer to page 102 for more information).

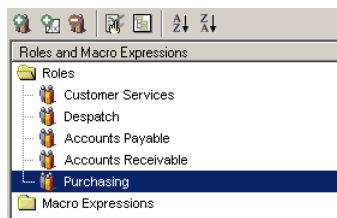



Figure 4.74: Adding a Role

4.9.1 Assigning a Role

Roles are assigned to specific Stages within a Process. Once assigned to a specific Stage, that Role is able to perform that specific Stage.

To assign a Role:

1. Click on the required Stage in the **Design for Mapping** window.
2. Click on the  Roles designer button.
3. Click and drag the required Role over the required Stage and release the mouse button. The Role has now been assigned to the Stage in a sub-folder called Roles.

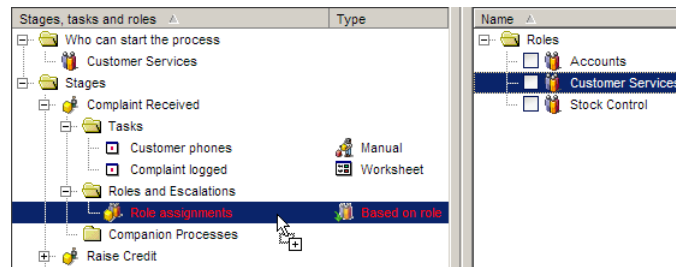


Figure 4.75: Assigning a Role

4. Repeat the procedure for multiple Stages, if required.

4.9.2 Filtering by Role

You can filter the model based on a particular Role. By choosing a particular Role you can see exactly what Stages that Role can perform.

To filter the model by Role:

1. Click on the **Processes** designer button in the **Design for Mapping** window.
2. Click on the **Roles** designer button.
3. Click on the tick box next to the Role that you want to highlight. The titles of all Processes and Stages that use that particular Role are now bolded.

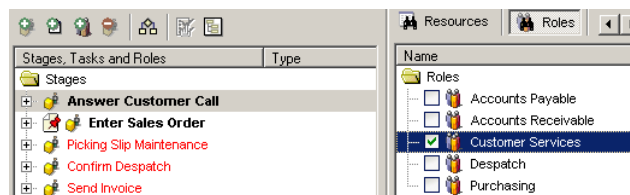


Figure 4.76: Filtering by Role

4.9.3 Changing a Role

Changing a Role involves adding or changing any properties for this item.

To change a Role:

1. Click on the required Role in the **Design Enterprise** or the **Design for Mapping** window.
2. Change any properties in the **Properties Editor**, if required.

4.9.4 Deleting a Role

You can delete a Role in the same way that you can delete any item in *XSOL Mapping* (refer to page 33 for more information). However, deleting a Role in different areas of the application does different things.

- **Design for Mapping** window - Deleting a Role here only deletes the Role assignment to a Stage in a Process. This removes the Role from the Process but not from the model itself. If you need to remove the Role from the entire model, you need to delete the Role in the **Design Enterprise** window.

- **Design Enterprise** window - Deleting a Role here deletes the Role from the entire model. You can only delete the Role if it is not being used in a Process.

4.10 Designing Worksheets



Automation Systems Design only

A worksheet is a form that allows a user to enter or maintain information in an XSOL system. There are two types of Worksheets, Task Worksheets and Maintenance Task Worksheets.

- **Task Worksheets** - A form that allows a user to enter or update information in an XSOL system. Task Worksheets are used in the execution of Tasks that require user input. You can have more than one Task Worksheet for each Task, useful for displaying different information depending on the Role of the user.

Before creating a Task Worksheet, you need to create the required Task as part of a Process and create the variables for all the information that you need the user to enter.

- **Maintenance Task Worksheets** - A form that allows a user to maintain table information in an *XSOL Automation* system.

Before creating a Maintenance Task Worksheet, you need to create the required Table and all the datafields that are in that Table.



- **Process Summary Worksheets** - A form that allows a user to view active and completed Process instances in an *XSOL Automation* system - refer to page 74 for more information.

Multiple Worksheets for the same item will appear as tabs in a single window in an *XSOL Automation* system. The list of Worksheets from top to bottom in *XSOL Mapping* corresponds to the tabbed Worksheets from left to right in *XSOL Automation*.








Worksheet properties:

- **Common Worksheet** - Common worksheets appear above all other Worksheets in an *XSOL Automation* system. If ticked, the Worksheet is a common Worksheet.
- **Panel Width** - Enter the required width for the Worksheet in the field followed by **p** (pixels), **m** (millimeters), **c** (centimeters), **i** (inches), or **p** (points).
- **Panel Height** - Enter the required height for the Worksheet in the field followed by **p** (pixels), **m** (millimeters), **c** (centimeters), **i** (inches), or **p** (points).
- **Default Label Alignment** - Select the standard label alignment used in the Worksheets. This alignment only applies to new widgets added to the Worksheets, and can be changed for each Widget, if required.
- **Reference Font Name** - Select the default font that is used in the Worksheet.
- **Reference Font Size in Pts** - Select the default font size that is used in the Worksheet.

To add a new Task or Maintenance Task Worksheet:

1. Make sure you have defined all the information that you require before creating a worksheet (refer to the description before this procedure for more information).
2. For Task Worksheets, do the following:
 - (a) Click on the  designer button in the **Design for Mapping** window.
 - (b) Click on the required Task.
 - (c) Click on the  button and enter the title of the Worksheet.

For Maintenance Task Worksheets, do the following:

- (a) Click on the  designer button in the **Design for Automation** window.
 - (b) Click on the  designer button.
 - (c) Click on the required Table.
 - (d) Click on the  button.
 - (e) Click on the  button and enter the name of the Worksheet, if required.
3. Common worksheets appear above all other Worksheets in an *XSOL Automation* system. If this is going to be the common Worksheet for this item, tick the **Common Worksheet** tick box in the **Properties Editor**.
-  You can always quickly turn the **Common Worksheet** setting on and off by selecting the Worksheet and clicking on the  button.
4. Click on the  button.

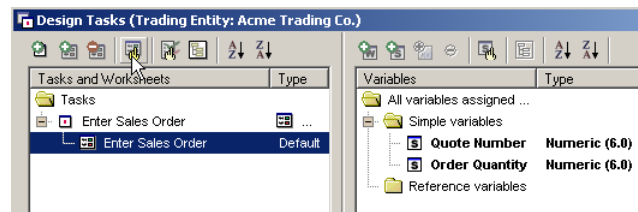



Figure 4.77: Opening the Design Worksheet Layout window

5. You are now in the **Design Worksheet Layout** window.
- If you are designing a new Task Worksheet, the work area will be blank. You need to create new widgets and/or drag and drop variables onto the worksheet to create data entry and display items. Refer to page 107 and page 110 for more information.
 - If you are designing a new Maintenance Task Worksheet, the work area will be pre-populated with a number of Input widgets that match the datafields in the table. You may need to change or delete widgets as required. Refer to page 107 and page 110 for more information.

4.10.1 Changing a Worksheet

 Automation Systems Design only

Changing a Worksheet can involve re-organizing the layout of the widgets on the form and adding or deleting widgets.

To change a Worksheet:

1. Open the **Design Worksheet Layout** window for the required Worksheet.
2. Change the widgets or layout of the worksheet as required. Refer to page 107 and page 110 for more information.



Figure 4.78: Changing a Worksheet

4.10.2 Using widgets



Automation Systems Design only

Widgets are graphical items that appear on a Worksheet that display data or store user input. Most widgets need to either get details from or store details in a Variable (which has its own data type about the sort of information it will display or store) and are automatically displayed in a format that is correct for the Variable's data type. There are also several additional widget types:


- **Label** - Displays a text label on the worksheet. Labels cannot be linked to a Variable.
- **Image** - Displays an image on the worksheet. An Image widget is not linked to a Variable, and only bitmap, JPEG, WMF, and EMF file types are supported. You can set the following type-specific fields:
 - **Url** - Insert the absolute path to the image file that you want to include. This can be a network or local path (e.g. **C:\images\logo.jpg**) or an Internet path (e.g. **http://www.mywebsite.com/images/logo.jpg**). Relative paths are not supported.
- You can use a Notational Variable instead of plain text. Use the following syntax - **xpp://VariableExternalName**. The external name for the Variable comes from the **External ID** property in the **Properties Editor**.
- **Alternate Text** - Insert a textual description of the image, used if the image is not available.
- **Image Repeat** - Select whether the image is repeated or not, and how it is repeated, from the drop-down list.
- **Bevel** - Adds a raised or sunken graphical item (line or box). Used to add visual information to the Worksheet, and has no effect on how the Worksheet works.
- **Grouping** - Displays a transparent labeled rectangular box. Used to visually group other widgets together, and has no effect on how the Worksheet works.
- **Horizontal Rule** - Displays a horizontal line. Used to visually separate widgets, and has no effect on how the Worksheet works.


To add a widget:

1. To create a widget from a Variable, drag and drop one or more Variables from the **Fields/Variables** list onto the Worksheet.



Figure 4.79: Adding a widget

To create a widget that does not require a Variable, select the widget type you want to create from the drop-down list next to the  button and click on the worksheet where you want to add the widget.

2. Click on a widget in the Worksheet and alter any properties in the **Properties Editor**, such as the caption or User Note information (refer to page 108 for more information), or enter the **Url**, **Width**, and **Height** properties if it is an Image widget.
3. Move the widget to a specific location on the Worksheet, if required (refer to page 110 for more information).
Refer to page 29 if you need to resize a widget. To return a widget to its default size (based on the **Appearance** property), click on the  button.
4. Repeat the last two steps for all the Variables that you have created.

Widget properties



Automation Systems Design only

Each widget has a large number of properties that are displayed in the **Properties Editor**. The following two lists describes fields that are common to most widgets, and fields that are specific to specific widget types.

General Properties:

- **Caption** - The name of the widget.
- **Widget Type** - Select the type of widget (refer to page 107 for a full list of widget types).
- **Top** - Indicates how far the widget is from the top of the Worksheet.
- **Left** - Indicates how far the widget is from the left side of the Worksheet.
- **Width** - Indicates the total width of the widget.
- **Height** - Indicates the total height of the widget.
- **Source** - Select the linked Variable for this widget.
- **Which occurrence** - Select which occurrence of the Variable to use, if there are multiple occurrences.
- **Inherit Label** - Use the caption of the linked Variable as the label for the widget.
- **Label alignment** - Select where the label appears in relation to the widget.
- **Label separation** - Enter how much distance is placed between a widget and its label. This overrides the default value of the chosen appearance.
- **Show indicator?** - Turn on or off the indicator next to the widget. Indicators are small graphics next to the field that provide additional information about the field. This overrides the default setting chosen with the **Appearance** property.
- **Purpose** - Select how the widget is used from the drop-down list. The following options are available:
 - **Display only** - Displays the field as read-only.
 - **Display with filter capability** - Display the field with an associated filter. Only available for Lookups.
 - **Mandatory input** - The field must be filled in order to process the Worksheet.
 - **Optional input** - The field can optionally be filled in. It is not required to process the Worksheet.

- **Appearance** - Select how the widget is displayed from the drop-down list. Each widget type has three different appearance options that display in a minimalist, compact, or full style.
- **Text Horizontal Alignment** - The horizontal text alignment of the label relative to the widget.
- **Text Vertical Alignment** - The vertical text alignment of the label relative to the widget.
- **Tab Index** - Define the order in which the focus is moved from widget to widget as the Tab key is pressed. The tab order starts at zero and increases in increments of one.
- **Hot key character** - Indicates the **Alt**-key sequence that moves focus to this widget. Enter a lower or upper case letter that exists in the label for the widget to select it as the **Alt** key.
- **No. of columns** - Specify the number of columns used to display the item (such as radio buttons for a Binary widget).

Specific Properties:

- **Bevel** - The following specific fields are available:
 - **Shape** - Select the shape of the bevel from the drop-down list.
 - **Style** - Select the style of the bevel, either sunken or raised.
- **Horizontal Rule** - Enter the thickness of the line, in pixels, in the **Thickness** field.
- **Image** - The following specific fields are available:
 - **URL** - Enter the location of the image file.
 - **Alternate Text** - Enter a textual description of the image. This displays when a cursor pauses over an image and is essential for blind web surfers who use web page reading software.
 - **Image Repeat** - Select if and how the image repeats on the Worksheet.
- **Label** - The following specific fields are available:
 - **Label for** - Select the variable that this label belongs to.

Changing a widget



Automation Systems Design only

Changing a widget can involve changing any of the widget's properties and altering its location on the Worksheet.

To change a widget:

1. Click on a widget in the **Design Worksheet Layout** window and alter any properties in the **Properties Editor**, such as the caption, User Note information, or change the widget type. Refer to page 108 for more information.
2. Move the widget to a specific location on the Worksheet, if required (refer to page 110 for more information).

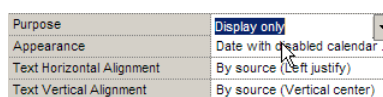


Figure 4.80: Changing a widget

4.10.3 Aligning widgets on a Worksheet



Automation Systems Design only

The **Design Worksheet Layout** window provides you with several functions that help you align and display widgets on a worksheet in a consistent manner. The following functions are available:

- - Displays a grid. The grid allows you to align items properly by using the grid as a visual frame of reference.
- - Snap to grid. Whenever a widget is moved on the Worksheet it will automatically align with the nearest grid markers. The small down arrow allows you to select the spacing between the grid markers.
- - Moves all selected widgets in line with the far left widget.
- - Moves all selected widgets in line with the far right widget.
- - Moves all selected widgets in line with the horizontal central point between all widgets.
- - Moves all selected widgets so that they are equally spaced horizontally.
- - Moves all selected widgets in line with the top widget.
- - Moves all selected widgets in line with the bottom widget.
- - Moves all selected widgets in line with the vertical central point between all widgets.
- - Moves all selected widgets so that they are equally spaced vertically.
- **Send to Back** - Places the selected item underneath any non-selected item that it overlaps.
- **Bring to Front** - Places the selected item in front of any non-selected item that it overlaps.

To align Worksheet widgets:

1. To turn the grid on or off, click on the button.
2. To enable or disable the Snap To Grid function, click on the button.
To set the size of the grid, click on the small down arrow next to the button.
3. To use any of the alignment buttons, select more than one widget (click and drag a box around the required widgets, or Ctrl+click on the required widgets), and click on the required button.

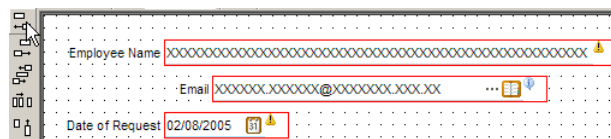


Figure 4.81: Aligning selected widgets horizontally

4. To place an item in front or behind of other items, right-click on the required item and click on the **Send to Back** or **Bring to Front** menu item.

4.10.4 Setting the Tab order on a Worksheet



Automation Systems Design only




The Tab order on a Worksheet is the order in which fields are focused on as the user presses the Tab key. A convenient arrangement of fields on a Worksheet, along with a Tab order that mirrors that arrangement, greatly increases the speed with which a user can enter information.

There are two ways you can set the Tab order:

1. **Tab Order Editor** - The Tab Order Editor allows you to set the Tab order for all or some of the Worksheet from top to bottom, or left to right, with one click. You can also drag and drop items to new positions, if required.
2. **Properties Editor** - You can manually set the **Tab Order** property for each item, as required. This is useful for quick changes, but not practical for changing an entire Worksheet.

To set the Tab order on a Worksheet:

1. To use the Tab Order Editor:

- (a) Click on the  button.
- (b) You are now in the **Tab Order** window. The Tab order list starts at the top of the list.
If you want to automatically order all the widgets as they appear from left to right on the Worksheet, click on the  button, or click on the  button to order only the selected widgets horizontally.

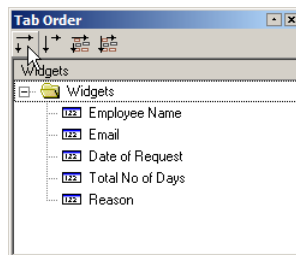





Figure 4.82: Setting the Tab order on a Worksheet

If you want to automatically order all the widgets as they appear from top to bottom on the Worksheet, click on the  button, or click on the  button to order only the selected widgets vertically.

If you want to manually rearrange the Tab order, drag and drop items within the list.

2. To use the Properties Editor:

- (a) Click on the widget that you want to change and open the Properties Editor, if it is not open already.
- (b) Change the order number in the **Tab Order** field.
 If you do not want the widget to appear in the Tab order at all, enter **-1** in the **Tab Order** field.
- (c) Repeat the above steps for as many widgets as required.

4.11 Designing Tables



Automation Systems Design only

There are several classifications of Tables within *XSOL Mapping*: Resources; Resource Extensions; Control Groups; and Relationships. These classifications are utilized to enforce certain principles of ESL and to simplify model construction. A Table is effectively the physical recording of logical groups of Data Fields within a database.

The Table classifications are utilized to enforce certain principles of ESL (Enterprise System Logic) and to simplify model construction.


There are several different groups of information within a Table:

- **Fields** - Fields, or datafields, are containers within a database record into which data is entered (refer to page 119 for more information).
- **Auto Display Fields** - Datafields that are automatically added to a Worksheet when that Table is added to a Worksheet (refer to page 105 for more information). Drag and drop the required datafields into this area in the middle pane to use this functionality.
- **Secondary Indices** - Datafields that help to optimize searching on a Table - refer to page 126 for more information.

Table properties:

- **Default Lookup** - Select the default Lookup List for this Table. This field will automatically choose the first Lookup List for this Table when it is created.

To design Tables:

1. Click on the Enterprise or Trading Entity that you require in the Hierarchy list in the **Design Enterprise** window.
2. Click on the  button.
3. You are now in the **Design for Automation** window. Refer to any of the following sections for more information about using Tables, or page 32 for more information about adding a Table Group.

4.11.1 Using Resources





Automation Systems Design only

The discrete things that an Enterprise employs when undertaking its business, such as Products, Customers, Staff and Bank Accounts. They are typically tangible assets that would appear on the balance sheet.

Resources have a sub-category called Resource Extensions (refer to page 113 for more information).

For an expanded definition please refer to page 20 for more information.

To add a Resource:

1. Click on the  button in the **Design for Automation** window.
2. Click on the  button.
3. Enter the name of the Resource and press the **Enter** key to create another item or press the **Tab** key to only create this item.



Do not use any of Python's reserved keywords as names, as they might cause an error in Python - refer to page 198 for more information.

4. Add one or more datafields. You need to add at least one datafield to the **Fields in UI** area in the middle pane to create a valid Resource. Refer to page 119 for more information about adding a datafield.
5. Add or change any properties in the **Properties Editor**, if required.

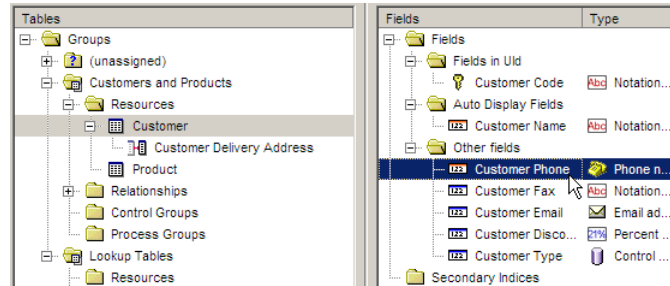


Figure 4.83: Adding a Resource


Changing a Resource



Automation Systems Design only

Changing a Resource involves adding or changing any properties for this item. Refer to page 123 if you want to change the Resource's datafields.

To change a Resource:

1. Click on the  **Tables** designer button in the **Design for Automation** window.
2. Click on the required Resource.
3. Change any properties in the **Properties Editor**, if required.
4. Change any datafields, if required.

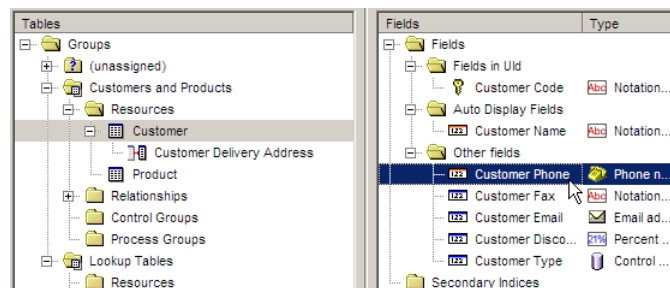


Figure 4.84: Changing a Resource

4.11.2 Using Resource Extensions






Automation Systems Design only

A logical group of Data Fields that is implicitly linked to a Resource but held as a separate Table. Each Resource may have many Resource Extensions and each Resource Extension may have a 1-to-1 or 1-to-many relationship to its parent Resource.

For an expanded definition please refer to page 20 for more information.

- Separate visibility
- Only add datafields to certain instances of the base Resource
- Database optimization

To add a Resource Extension:

1. Click on the  **Tables** designer button in the **Design for Automation** window.
 2. Click on the Resource to which you need to add the Resource Extension.
 3. Click on the  button.
 4. Enter the name of the Resource Extension and press the **Enter** key to create another item or press the **Tab** key to only create this item.
-  Do not use any of Python's reserved keywords as names, as they might cause an error in Python - refer to page 198 for more information.
5. Add one or more datafields. You do not need to add any datafields to create a valid Resource Extension. Refer to page 119 for more information about adding a datafield.
 6. Add or change any properties in the **Properties Editor**, if required.

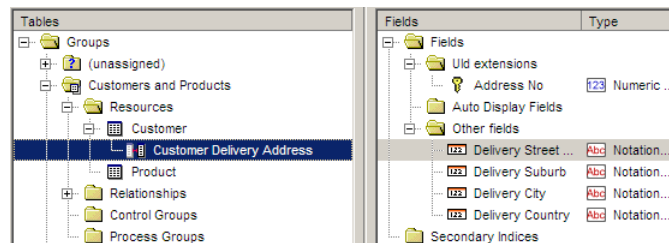


Figure 4.85: Adding a Resource Extension


Changing a Resource Extension



Automation Systems Design only

Changing a Resource Extension involves adding or changing any properties for this item. Refer to page 123 if you want to change the Resource Extension's datafields.

To change a Resource Extension:

1. Click on the  **Tables** designer button in the **Design for Automation** window.
2. Click on the required Resource Extension.
3. Change any properties in the **Properties Editor**, if required.

4. Change any datafields, if required.

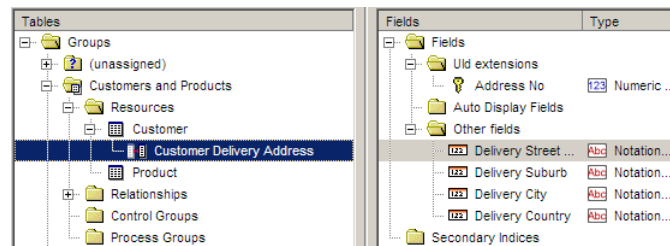


Figure 4.86: Changing a Resource Extension

4.11.3 Using Control Groups



Automation Systems Design only

A Table that usually comprises a code and description that is used to validate input content on other Tables and then enable grouping of data elements for viewing and/or reporting. These are typically validation- or grouping-type Tables.

For an expanded definition please refer to page 21 for more information.

Control Group properties

- **Model Defined Entries** - The data for this Control Group are defined in the model - refer to page 116 for more information.

To add a Control Group:

1. Click on the **Tables** designer button in the **Design for Automation** window.
2. Click on the button.
3. Enter the name of the Control Group and press the **Enter** key to create another item or press the **Tab** key to only create this item.
- ⚠ Do not use any of Python's reserved keywords as names, as they might cause an error in Python - refer to page 198 for more information.
4. Add one or more datafields. You need to add at least one datafield in the **Fields in UId** area and at least one datafield in the **Other fields** area (both in the middle pane) to create a valid Control Group. Refer to page 119 for more information about adding a datafield.
5. Add or change any properties in the **Properties Editor**, if required.

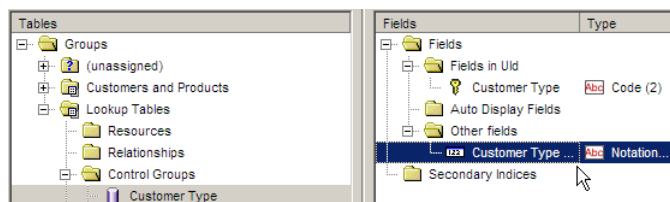


Figure 4.87: Adding a Control Group

Using model defined Control Groups



Automation Systems Design only

You can manually add Control Group data to Control Groups in a model. Model defined Control Group entries are useful when you need to have specific data within a Control Group, as these entries are contained within the model itself, not in the SQL database used by *XSOL Automation*. These entries should only be used or values that very rarely change, as you need to re-commit the model to alter these values.




A Control Group Table with one or more entries can not have a Maintenance Task assigned to it (refer to page 105 for more information).



Model defined Control Groups entries are optional. If not used data must be entered into the *XSOL Automation* SQL database in the same manner as all other Tables.

To add entries to a model defined Control Group:

1. Click on the  **Tables** designer button in the **Design for Automation** window.
2. Click on the required Control Group in the left pane.
3. Click on the **Model Defined Entries** property in the **Properties Editor**.



You can not do this if the Control Group has one or more Maintenance Tasks assigned to it. All Maintenance Tasks must be deleted before you can enable this property - refer to page 105 for more information.

4. Refer to page 116 to add entries to this Control Group.


Adding or editing model defined Control Group entries



Automation Systems Design only

You can add or edit model defined Control Group entries at any time - refer to page 116 for more information.

To add or edit a model defined Control Group entry

1. Click on the  **Control group entries** designer button in the **Design for Automation** window.
2. Click on the required Control Group in the left pane.
3. To add an entry:
 - (a) Click on the first blank field in the **Code** column. Enter the entry code in this field.
 - (b) Enter description information in the adjacent column. This information is optional.
 - (c) Repeat for as many Control Group entries as required.


Code 	Description
M	Male
F	Female
X	
*	

Figure 4.88: Adding a Control Group entry

To edit an entry:

- (a) Click on the required **Code** or **Description** column entry and edit as required.
- (b) Repeat for as many Control Group entries as required.



Deleting a model defined Control Group entry



Automation Systems Design only

You can delete model defined Control Group entries at any time - refer to page 116 for more information.

To delete a model defined Control Group entry:

1. Click on the  **Control group entries** designer button in the **Design for Automation** window.
2. Click on the required Control Group in the left pane.
3. Click on the required Control Group entry and click on the  button.
4. Click on the **Yes** button in the confirmation window.


Changing a Control Group



Automation Systems Design only

Changing a Control Group involves adding or changing any properties for this item. Refer to page 123 if you want to change the Control Group's datafields.

To change a Control Group:

1. Click on the  **Tables** designer button in the **Design for Automation** window.
2. Click on the required Control Group.
3. Change any properties in the **Properties Editor**, if required.
4. Change any datafields, if required.

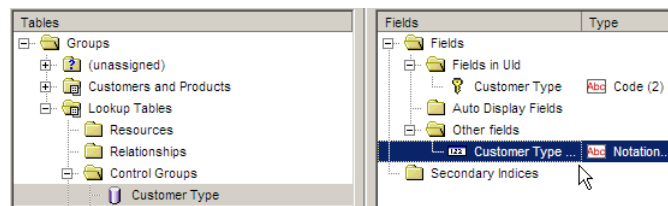


Figure 4.89: Changing a Control Group

4.11.4 Using Relationships





Automation Systems Design only

A Table used to hold information that is relevant to one or more Resources and/or Control Groups that can also have optional range parameters specified. A Relationship has three separate segments in its Uid: Tables; Ranges; and Datafields. A Relationship must contain a minimum of one Resource or Control Group but can contain more of either table type.



The range parameters are optional and include date, time, datetime (mutually exclusive with the date and time options), days of the week, and the mutually exclusive numeric and percentage breakpoints.

To add a Relationship:

1. Click on the  designer button in the **Design for Automation** window.
2. Click on the  button.
3. Enter the name of the Relationship and press the **Enter** key to create another item or press the **Tab** key to only create this item.



Do not use any of Python's reserved keywords as names, as they might cause an error in Python - refer to page 198 for more information.

4. Drag and drop one or more Tables from the  area to the **Tables in UId** folder in the middle pane. You need to add at least one Table in the middle pane to create a valid Relationship.
5. Add one or more datafields, as required. Refer to page 119 for more information about adding a datafield.
6. Click on the **Ranges in UId** folder in the middle pane to set one or more ranges for the Relationship.
7. You now need to set up one or more range types. Ranges are the potential limits on relationships between datafields - users set up the actual limits in *XSOL Automation*. Click on one of the options from the  menu button and edit the following options in the **Properties Editor**:



The Date and Time ranges, and the Date/Time range, are mutually exclusive, as are the Numeric and Percent ranges.

- **Date Range** - Allows *XSOL Automation* users to define one or more date ranges, such as from 01/01/2004 to 31/12/2004. Tick the **Start** and/or **End** tick boxes to allow users to define start or end values for the range.
- **Time Range** - Allows *XSOL Automation* users to define one or more time ranges, such as from 09:00am to 5.30pm. Tick the **Start** and/or **End** tick boxes to allow users to define start or end values for the range.
- **Date and Time Range** - Allows *XSOL Automation* users define one or more time ranges within a date range, such as from 09:00am to 5.30pm on each day from 01/01/2004 to 31/12/2004. Tick the **Start** and/or **End** tick boxes to allow users to define start or end values for each range.
- **Datetime Range** - Allows *XSOL Automation* users to define one or more date/time ranges, such as from 09:00am 01/01/2004 to 5.30pm 31/12/2004. Tick the **Start** and/or **End** tick boxes to allow users to define start or end values for the range.
- **Days Of The Week Selection** - Allows *XSOL Automation* users to define one or more ranges based on weekdays, such as Monday, Wednesday, and Friday. Tick any of the weekday tick boxes to allow the days to be used.
- **Numeric Breakpoint Range** - Allows *XSOL Automation* users to define one or more numeric ranges, such as 10-20 or 50-100. Tick the **Start** and/or **End** tick boxes to allow users to define start or end values for the range.
- **Percent Breakpoint Range** - Allows *XSOL Automation* users to define one or more percentage ranges, such as 5%-15% or 85%-95%. Tick the **Start** and/or **End** tick boxes to allow users to define start or end values for the range.



XSOL Automation chooses the most finite range if two or more ranges overlap. For example, 01/01/2004 to 31/12/2004 and 01/04/2004 to 30/04/2004 have both been defined. Both ranges apply, but the second is chosen in that date period.

8. Tick the **Allow overlapping date/time range values?** tick box for the **Ranges in UId** folder if you want date, time, or date/time range values to overlap in *XSOL Automation*.

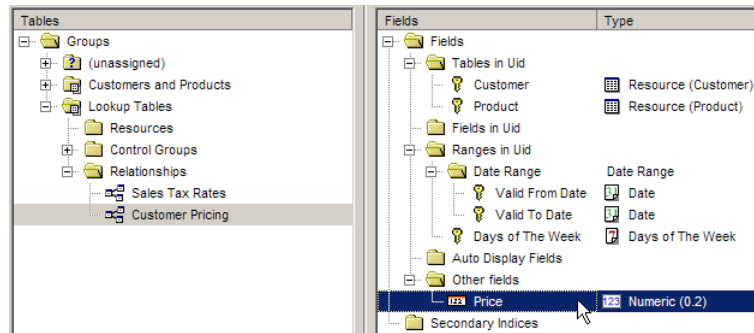


Figure 4.90: Adding a Relationship

Changing a Relationship



Automation Systems Design only

Changing a Relationship involves adding or changing any properties for this item. Refer to page 123 if you want to change the Relationship's datafields.

To change a Relationship:

1. Click on the **Tables** designer button in the **Design for Automation** window.
2. Click on the required Relationship.
3. Change any properties in the **Properties Editor**, if required.
4. Change any datafields, if required.

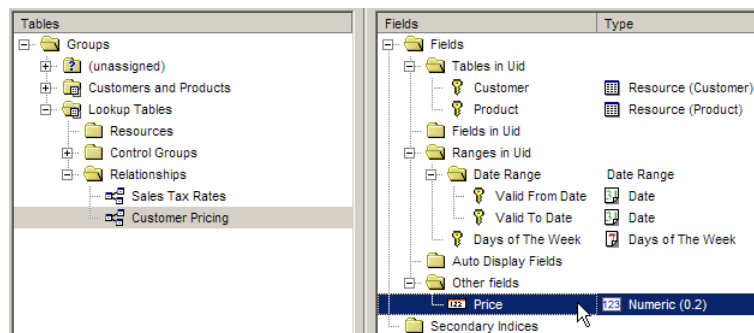


Figure 4.91: Changing a Relationship

4.11.5 Using datafields



Automation Systems Design only

Datafields are containers within a database record into which data is entered. Depending on the type of Table that the datafield is created in, there may be specific information that needs to be entered in the

datafield. The two mandatory Property fields that need to be filled in for each datafield are the **Caption** and **Data Type** fields.

The following list describes fields in the **Properties Editor** that are common to most data types.

General Properties:

- **Caption** - The name of the datafield.
- **Data Type** - The type of data used in this datafield. These are described in more detail below.
- **Mandatory** - Indicates whether the datafield is mandatory or not. mandatory datafields must have an entered value when used on a Worksheet or assigned to a Task.
- **Repeat Count** - The number of fields that are used for this datafield. For example, a Repeat Count of four will create four separate fields within the same datafield.
- **Datafield default** - The default value that is displayed in this datafield. Enter either a fixed value or an Expression.
- **Calculated** - Indicates whether the datafield is calculated or not. When added to a Worksheet, a calculated datafield calculates the Expression added in the **Calculation** property when the Worksheet is opened or when a Variable in the Expression is updated.
- **Calculation** - Add an Expression that is executed when a Worksheet with this datafield on it is opened or when a Variable in the Expression is updated. Only available when the **Calculated** tick box is checked.
- **Input Mask** - Restrictions on the type of data that can be entered into a datafield. For example, a Notational field can automatically enter Male, Female, or X depending on whether the user types M, F, or X, and no other input is allowed. Refer to page 123 for more information.
- **Display Mask** - Restrictions on how data in a datafield is displayed. For example, numeric data can be displayed using a thousands separator (1,000 instead of 1000). Refer to page 125 for more information.
- **User Notes** - Allows you to enter various types of additional information about the datafield. Refer to page 145 for more information.
- **External ID** - The name of the datafield, displayed in the Caption field, may not be compatible with external databases when the system is run. For example, SQL does not allow spaces in data names. The XPM (XSOL Process Manager) will automatically convert the Caption to a usable name and insert that name into the External ID field. You can change this name, if required, but you can only enter allowed names into this field, for example no spaces in the name. This value is also used when running XML (Extensible Markup Language) reports.
- **Run-time Hint** - A small piece of information about this datafield that can help a user in the run-time system. This can appear as text in the status bar or displayed in a window.

Data Type descriptions:

- **Auto Increment** - Used for identifying and/or grouping Process Events.

You can set the following type-specific fields:

- Start At - Starting sequence number, which cannot be zero. Mandatory.
- End At - Optional ending sequence number.
- Step - The incremental step, which cannot be zero. Mandatory.



Auto Increment fields can not be rolled back in an *XSOL Automation* live system - once a number is used it is never used again by that field. The best place to put an Auto Increment value is in a system Task (such as **Create Document from a Template** or **System Calculation**) in a system Stage (a Stage that only contains system Tasks), as this can not be aborted by a user. This means there will not be any gaps in the Auto Increment numbering.



- **Binary** - Used when only two value are allowed (also known as Boolean). You can set the following type-specific fields:
 - True Value - The true value of the field. Mandatory.
 - False Value - The false value of the field. Mandatory.
 - Null or unknown value - The text to display if the field has no value. Optional.
- **Code** - Notational field that identifies the data type as a Control Group. Mandatory for a Control Group.
- **Control Group** - Refers to a specific Control Group. You can set the following type-specific field:
 - Control Group - Choose the required Control Group. Mandatory.
- **Date** - Used to display dates. You can set date formats using the Input Mask and Display Mask fields.
- **Datetime** - Used to display dates and times in one field. You can set date and time formats using the Input Mask and Display Mask fields.
- **Document** - Used for manipulating documents. You can set the following type-specific fields:
 - **Document Location** - Click on the required Document Location from the drop-down list.
 - **Existence Type** - Choose from the drop-down list whether the document must or must not exist.

You need to add one or more Business Rules to the Variable if you need to allow Users to select documents in an *XSOL Automation* system (refer to page 67 for more information).


- **Email Address** - Used for entering and manipulating email addresses.
- **Index** - Used as an index for another table. You can set the following type-specific field:
 - Datafield - Choose the required Datafield from another Table. Mandatory.
- **Memo** - Used for entering general alphanumeric characters. Memo fields can span multiple lines of text (unlike Notational fields) and are limited to 4000 characters.
- **Notational** - Used for entering general alphanumeric characters. Notational fields are limited to 120 characters in length and one line of text - multiple lines are not allowed. You can set the following type-specific field:
 - Length - The length, in characters, of the field. Mandatory.
- **Numeric** - Used for entering numeric characters. You can set the following type-specific fields:
 - No of Integer Digits - How many digits are allowed before the decimal place. Mandatory.
 - No of Decimal Digits - How many digits are allowed after the decimal place. Mandatory.
 - Allow Zero - Indicate whether zero is an acceptable value.
 - Allow Positive - Indicate whether positive numbers are acceptable values.
 - Allow Negative - Indicate whether negative numbers are acceptable values.

- Ranges - Specify what range of numbers are acceptable values. Use two periods (..) to indicate a range between two numbers, and commas (,) to separate ranges. For example, **-1..10,12..50** will allow all numbers between -1 and 10, and between 12 and 50.
- **Parent Lookup** - Refers to another Resource, which enables you to create a hierarchy of Resources. Only available for Resource fields.
- **Percent** - Used for entering percentile information. You can set the following type-specific fields:
 - No of Decimal Digits - How many digits are allowed after the decimal place. Mandatory.
 - Allow Zero - Indicate whether zero is an acceptable value.
 - Allow Positive - Indicate whether positive numbers are acceptable values.
 - Allow Negative - Indicate whether negative numbers are acceptable values.
 - Ranges - Specify what range of numbers are acceptable values. Use two periods (..) to indicate a range between two numbers, and commas (,) to separate ranges. For example, **-1..10,12..50** will allow all numbers between -1 and 10, and between 12 and 50.
- **Phone Number** - Used for entering and manipulating phone numbers. You can set the following type-specific field:
 - Country Code Mandatory - Indicate whether country code information is required for this field.
- **Printer** - Used for storing printer information. You can set the following type-specific field:
 - Default Logical Printer - Choose the printer used by this field.
- **Scientific** - Used for floating point numeric values. For example, use a Scientific datafield when users need to enter numeric information which has an unknown number of decimal places.
- **Time** - Used to display time values. You can set time formats using the Input Mask and Display Mask fields.
- **User** - Used to choose a particular user at run-time. The data type allows you to add a Business Rule to which you can assign a Role. In *XSOL Automation* you get a drop-down list that lists all the users available for those Roles (refer to page 132 for more information).

To add a datafield:

1. Click on the  **Tables** designer button in the **Design for Automation** window.
2. Click on the required Table.
3. Click on the  button.

There are a variety of specific add buttons for different Tables:

-  - Create Control Group Field (Control Group)
4. Enter the name of the datafield and press the **Enter** key to create another item or press the **Tab** key to only create this item.
 5. Click on the data type for the datafield from the **Data Type** field. Enter any specific information for this data type in the **Properties Editor**.

6. Add or change any properties in the **Properties Editor**, if required.

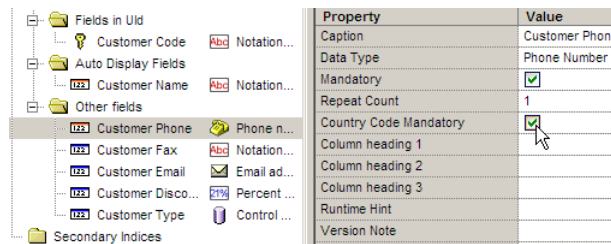


Figure 4.92: Adding a datafield


Changing a datafield



Automation Systems Design only

Changing a datafield involves adding or changing any properties for this item. Refer to page 119 for more information about specific properties of various data types.

To change a datafield:

1. Click on the  designer button in the **Design for Automation** window.
2. Click on the required datafield.
3. Change the properties or any User Notes in the **Properties Editor**, if required.

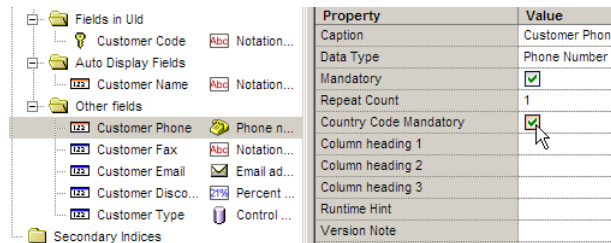


Figure 4.93: Changing a datafield

Input Mask syntax



Automation Systems Design only

Input Masks limit the input that an *XSOL Automation* user can enter into a field.



Display Masks can be added both to Variables and Widgets. If used in both areas, the Display Mask on the Widget takes precedence.

Examples

- ([Mm] ale)|([Ff] emale)|([Xx]) - Allows users to enter M, F, or X (upper or lower case) and get Male, Female, or X.
- ([Yy] es)|([Nn] o) - Allows users to enter Y, or N (upper or lower case) and get Yes or No.

- `$[0-9]*[.]?[0-9]?[0-9]` - Allows users to enter currency values. Starting with a dollar sign, users can enter any number of digits, optionally be followed by two decimal digits. A decimal point must be entered between the integer and decimal digits.
- `0?[0-9]|1[0-9]|2[0-3]` - Allows users to enter any number between zero and twenty-three (such as 11 or 15, but not 35).
- `[0-9]{2-[0-9]{4`
`-[0-9]{7-[0-9]{3}}` - Allows users to enter a bank account number in this format: `xx-xxxx-xxxxxxx-xxx`.

XSOL Automation uses standard Regular Expressions syntax for Input Masks. A concise summary of the syntax used is given below:

- `.` - Matches any symbol. It is treated as full stop within square brackets.
- `*` - Matches the preceding expression zero or more times.
- `+` - Matches the preceding expression one or more times.
- `?` - Matches the preceding expression zero or one times.
- `{n}` - Matches the preceding expression exactly n times.
- `()` - Character sequence grouping. Matches any symbol in the sequence in which they are listed, any number of times. `(a)+` matches a, aaa, or aaaa.
- `[]` Grouping. Any symbol listed within square brackets can represent a character. `[bfc]at` matches bat, fat and cat.
- `[^]` - Exclusive grouping. Any symbol except those in square brackets can represent a character. `[^abc]` matches any symbol except a, b, and c.
- `(|)` - Variant grouping. Any of two or more sequences can match. `(b)oo` matches both zoo and boo.
- `\` - Escape character. Allows any special characters to be used as a regular character.
- `\d` - Matches a digit character. Equivalent to `[0-9]`.
- `\D` - Matches a non digit character. Equivalent to `[^0-9]`.
- `\f` - Matches a form-feed character. Equivalent to `\x0c`.
- `\n` - Matches a new line character. Equivalent to `\x0a`.
- `\r` - Matches a carriage return character. Equivalent to `\x0d`.
- `\s` - Matches any white space character including space, tab, form-feed, etc.
- `\S` - Matches any non white space character.
- `\t` - Matches a tab character. Equivalent to `\x09`.
- `\w` - Matches any word character including underscore. Equivalent to `'[A-Za-z0-9_]'`.
- `\W` - Matches any non-word character. Equivalent to `'[^A-Za-z0-9_]'`.
- `\xn` - Matches n, where n is a hexadecimal escape value. Hexadecimal escape values must be exactly two digits long. This allows ASCII codes to be used in Regular Expressions.

Using a Display Mask

Display Masks limit the display that an *XSOL Mapping* user sees in a User Note or an *XSOL Automation* user sees in a field. Display Masks can only be used on arithmetic, date, datetime, numeric, percent, scientific, and time Variables.



For *XSOL Automation*, display Masks can be added both to Variables and Widgets. If used in both areas, the Display Mask on the Widget takes precedence.



Display Masks do not alter a stored value, only display it according to specified rules. This can cause confusion if used inappropriately with the **No of Integer Digits** and **No of Decimal Digits** properties.


A summary of the syntax used is given below:

- 0 - Digit placeholder. Either a digit or a "0" is stored in the corresponding position in the output.
- # - Digit placeholder. Either a digit or nothing is stored in the corresponding position in the output.
- . - Decimal point. Used to indicate whether the decimal point is in the output.
- , - Thousand separator. If the format string contains one or more "," characters, the output will have thousand separators inserted between each group of three digits to the left of the decimal point. The placement and number of "," characters in the format string does not affect the output, except to indicate that thousand separators are wanted.
- E+ / E- - Scientific notation. Indicates that the output number is displayed using scientific notation.
- 'xx'/"xx" - Characters enclosed in single or double quotes are displayed as-is, and do not affect any other formatting.

Examples

- 0000 - Using a test value of 999 this displays 0999. Note the extra zero.
- #.## - Using a test value of 12.2 this displays 12.2. Note the extra digit to the left of the decimal still appears.
- '\$'#.00 - Using a test value of 2.5 this displays \$2.50. Note the dollar sign and the extra zero. The field will always show two decimal places.
- 00.## - Using a test value of .006 this displays 00.01. Note the extra zeros to the right of the decimal point and the rounding to two decimal places.

To create a Display Mask:

1. Click on the User Note or Variable to which you want to add a Display Mask.
2. Click on the **Display Mask** property in the **Properties Editor** and click on the  button.
3. Enter the Display Mask values for positive, negative, and zero values in the **Positive**, **Negative**, and **Zero** fields. All fields are optional.
4. Enter a value in the **Test** field and click on the **Test** button to test the Display Mask.

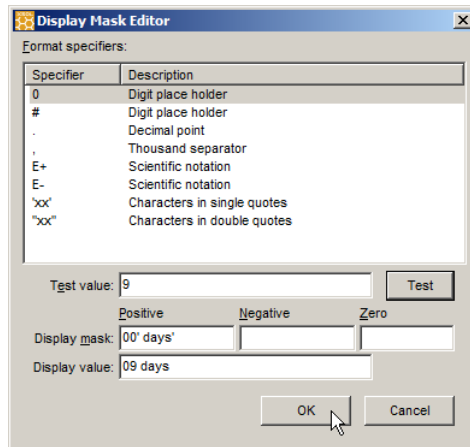


Figure 4.94: Creating a Display Mask

- Click on the **OK** button to save the Display Mask.

4.11.6 Using Secondary Indices



Automation Systems Design only

Secondary Indices help to optimize searching on a Table. They are generally used in two different ways:

- Limit a field, which you do not want to use as a UId field, to unique values only. For example, you may have a "Customer" Resource with a "Customer ID" UId field and "Customer Name" as a regular field. If you add "Customer Name" to a unique Secondary Index, you can only enter "Acme" in the "Customer Name" field once, whereas if you made "Customer Name" a UId field you could enter it multiple times, as long as the "Customer ID" field was different each time.
- Add a Business Rule to a non-UId field. Business Rules on a Table can normally only be added to a UId field. However you can create a Secondary Index for that field and add a Business Rule to it that way - refer to page 134 for more information.

To add a Secondary Index:

- Click on the **Tables** designer button in the **Design for Automation** window.
- Click on the required Table.
- Click on the button.
- Enter the name of the Secondary Index and press the **Enter** key to create another item or press the **Tab** key to only create this item.
- De-select the **Index** property in the **Properties Editor** if you do not want to restrict the datafield to unique values.
- Drag and drop any of field in the Table onto the Secondary Index to add it. Repeat as required.

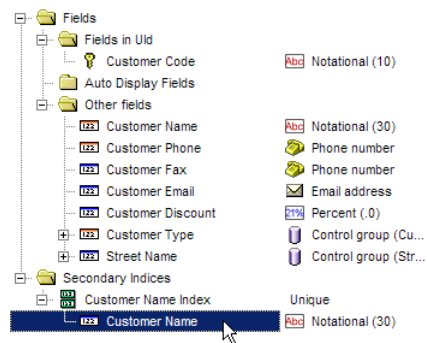


Figure 4.95: Adding a Secondary Index

Changing a Secondary Index



Automation Systems Design only

Changing a Secondary Index involves changing any properties for this item and adding or removing datafields.

To change a Secondary Index:

1. Click on the **Tables** designer button in the **Design for Automation** window.
2. Click on the required Table.
3. Click on the required Secondary Index in the middle pane.
4. Change any properties in the **Properties Editor**, if required.
5. Change any datafields, if required.

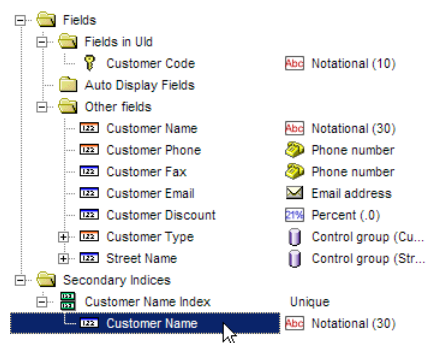


Figure 4.96: Changing a Secondary Index


4.12 Designing Variables



Automation Systems Design only

Variables are containers for holding data that is input through a Process. For example, in the Expression **Price * Quantity** both Price and Quantity are Variables. Variables have their own data type, much like datafields, which dictates what sort of information can be entered into them. Variables are used in Worksheets to accept data input by a user or to store information resulting from a calculation or Expression.

There are several different types of Variables:


- **Process Variables** Variables used in the executions of Processes.
 - **Stored** - A permanent Variable that is stored in the event output from a Task. Stored Variables are used for data entry into Worksheets or to hold the result of a calculation or Expression, and are available for any other Tasks in the Process.
 - **Work** - A temporary Variable that holds information only for the life of the Task. The information can be from data entry or a result of a calculation or Expression. Work Variables should be used for any information that can be calculated from other Variables, as this keeps the model and database simpler.
-  In Tasks performed by the system, only Work Variables can have an Expression assigned, i.e. if you need to run an Expression on a system Task, you need to use a Work Variable.
- **Reference** - Selects an instance of a Table. Refer to page 133 for more information.
- **Datafield** - Enables a datafield to be modifiable in a Worksheet. Refer to page 135 for more information.

Variable data types are the same as datafield data types - refer to page 119 for more information.

Variables often have an Expression associated with them (refer to page 138 for more information). Expressions that are associated with a Variable can have two states:

- **Assigned** - The Expression runs when the Variable is Processed.
- **Not Assigned** - The Expression does not run when the Variables is Processed.

To design a Variable:

1. Click on the Enterprise or Trading Entity that you require in the Hierarchy list in the **Design Enterprise** window.
2. Click on the  button.
3. Refer to any of the following sections for more information about using Variables, or page 32 for more information about adding a Variable Group.

4.12.1 Using Process Variables



Automation Systems Design only

There are two types of Process Variables:

- **Stored** - A permanent Variable that is stored in the event output from a Task. Stored Variables are used for data entry into Worksheets or to hold the result of a calculation or Expression, and are available for any other Tasks in the Process.

- **Work** - A temporary Variable that holds information only for the life of the Task. The information can be from data entry or a result of a calculation or Expression. Work Variables should be used for any information that can be calculated from other Variables, as this keeps the model and database simpler.



In Tasks performed by the system, only Work Variables can have an Expression assigned, i.e. if you need to run an Expression on a system Task, you need to use a Work Variable.

General Properties:

- **Caption** - The name of the datafield.
- **Data Type** - The type of data used in this datafield. These are described in more detail below.
- **Store variable** - If ticked indicates that this is a Stored Variable. If not ticked indicates that this is a Work Variable.
- **Variable default** - The default value that is displayed in this Variable. Enter either a fixed value or an Expression.

Default values are set at the beginning of the Process for Stored Variables, and at the beginning of the Task for Work Variables. A default value set on a mandatory field in a Worksheet means that the *XSOL Automation* user does not have to alter the field to successfully complete the Worksheet. For example, a default value applied to a mandatory Binary field means that the Binary field is automatically correct without the *XSOL Automation* user altering the field.



If a default value and a non-default Expression are both applied to a Variable, the non-default Expression takes precedence.

- **Input Mask** - Restrictions on the type of data that can be entered into a datafield. For example, a Date field may only allow data entry in the format dd/mm/yyyy.
- **Display Mask** - Restrictions on how data in a datafield is displayed. Essentially the same as Input Mask, except for display only.
- **Repeat Count** - The number of fields that are used for this datafield. For example, a Repeat Count of four will create four separate fields within the same datafield.
- **User Notes** - Allows you to enter various types of additional information about the datafield. Refer to page 145 for more information.
- **External ID** - The name of the datafield, displayed in the Caption field, may not be compatible with external databases when the system is run. For example, SQL does not allow spaces in data names. The XPM (XSOL Process Manager) will automatically convert the Caption to a usable name and insert that name into the External ID field. You can change this name, if required, but you can only enter allowed names into this field, for example no spaces in the name. This value is also used when running XML (Extensible Markup Language) reports.
- **Run-time Hint** - A small piece of information about this datafield that can help a user in the run-time system. This can appear as text in the status bar or displayed in a window.


Data Type descriptions:



- **Arithmetic** - Used for entering numeric characters that usually specify a quantity (of stock, for example). You can set the following type-specific fields:
 - No of Integer Digits - How many digits are allowed before the decimal place. Mandatory.

- No of Decimal Digits - How many digits are allowed after the decimal place. Mandatory.
 - Allow Zero - Indicate whether zero is an acceptable value.
 - Allow Positive - Indicate whether positive numbers are acceptable values.
 - Allow Negative - Indicate whether negative numbers are acceptable values.
 - Ranges - Specify what range of numbers are acceptable values. Use two periods (..) to indicate a range between two numbers, and commas (,) to separate ranges. For example, **-1..10,12..50** will allow all numbers between -1 and 10, and between 12 and 50.
 - **Auto Increment** - Used for identifying and/or grouping Process Events. You can set the following type-specific fields:
 - Start At - Starting sequence number, which cannot be zero. Mandatory.
 - End At - Optional ending sequence number.
 - Step - The incremental step, which cannot be zero. Mandatory.
 - **Binary** - Used when only two value are allowed (also known as Boolean). You can set the following type-specific fields:
 - True Value - The true value of the field. Mandatory.
 - False Value - The false value of the field. Mandatory.
 - Null or unknown value - The text to display if the field has no value. Optional.
 - **Date** - Used to display dates. You can set date formats using the Input Mask and Display Mask fields. You can set the following type-specific field:
 - Range Type - Defines how this datafield behaves with other datafields. The following options are available:
 - * **Absolute** - Use the stated date only.
 - * **End** - Use the stated date as the end of a date range.
 - * **Start** - Use the stated date as the start of a date range.
 - **Datetime** - Used to display dates and times in one field. You can set date and time formats using the Input Mask and Display Mask fields. You can set the following type-specific field:
 - Range Type - Defines how this datafield behaves with other datafields. The following options are available:
 - * **Absolute** - Use the stated date only.
 - * **End** - Use the stated date as the end of a date range.
 - * **Start** - Use the stated date as the start of a date range.
 - **Days of the Week** - Used for selecting one or more week days for date/time ranges. You can set the following type-specific fields:
 - Days of the week - Select one or more of the weekday tick boxes.
 - **Document** - Used for manipulating documents. You can set the following type-specific fields:
 - **Document Location** - Click on the required Document Location from the drop-down list.
 - **Existence Type** - Choose from the drop-down list whether the document must or must not exist.
 - **Email Address** - Used for entering and manipulating email addresses.
-

- **Memo** - Used for entering general alphanumeric characters. Memo fields are not limited in length and can span multiple lines of text (unlike Notational fields).
- **Notational** - Used for entering general alphanumeric characters. Notational fields are limited to 120 characters in length and one line of text - multiple lines are not allowed. You can set the following type-specific field:
 - Length - The length, in characters, of the field. Mandatory.
- **Numeric** - Used for entering numeric characters. You can set the following type-specific fields:
 - No of Integer Digits - How many digits are allowed before the decimal place. Mandatory.
 - No of Decimal Digits - How many digits are allowed after the decimal place. Mandatory.
 - Allow Zero - Indicate whether zero is an acceptable value.
 - Allow Positive - Indicate whether positive numbers are acceptable values.
 - Allow Negative - Indicate whether negative numbers are acceptable values.
 - Ranges - Specify what range of numbers are acceptable values. Use two periods (..) to indicate a range between two numbers, and commas (,) to separate ranges. For example, **-1..10,12..50** will allow all numbers between -1 and 10, and between 12 and 50.
- **Percent** - Used for entering percentile information. You can set the following type-specific fields:
 - No of Decimal Digits - How many digits are allowed after the decimal place. Mandatory.
 - Allow Zero - Indicate whether zero is an acceptable value.
 - Allow Positive - Indicate whether positive numbers are acceptable values.
 - Allow Negative - Indicate whether negative numbers are acceptable values.
 - Ranges - Specify what range of numbers are acceptable values. Use two periods (..) to indicate a range between two numbers, and commas (,) to separate ranges. For example, **-1..10,12..50** will allow all numbers between -1 and 10, and between 12 and 50.
- **Phone Number** - Used for entering and manipulating phone numbers. You can set the following type-specific field:
 - Country Code Mandatory - Indicate whether country code information is required for this field.
- **Printer** - Used for storing printer information. You can set the following type-specific field:
 - Default Logical Printer - Choose the printer used by this field.
- **Scientific** - Used for numeric values that may vary in size. For example, use a Scientific datafield when users need to enter numeric information which has an unknown number of decimal places.
- **Time** - Used to display time values. You can set time formats using the Input Mask and Display Mask fields.
- **User** - Used to choose a particular user at run-time. The data type allows you to add a Business Rule to which you can assign a Role. In *XSOL Automation* you get a drop-down list that lists all the users available for those Roles (refer to page 132 for more information).

To add a Stored or Work Variable:

1. Click on the  Variables designer button in the **Design for Automation** window.
2. Click on the required Variable Group to which you want to add the new Variable.

- Click on the  button to add a Stored Variable or the  button to add a Work Variable.
- Enter the name of the Variable and press the **Enter** key to create another item or press the **Tab** key to only create this item.



Do not use any of Python's reserved keywords as names, as they might cause an error in Python - refer to page 198 for more information.

- Click on the data type for the Variable from the **Data Type** field. Enter any specific information for this data type in the **Properties Editor**.
- Add or change any properties in the **Properties Editor**, if required.

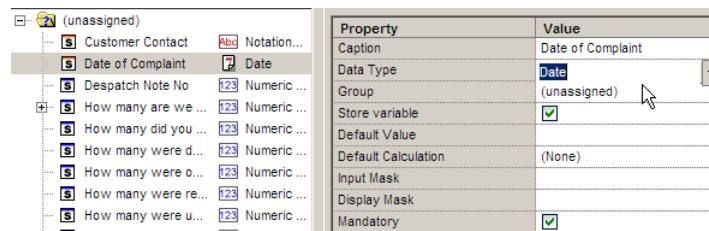


Figure 4.97: Adding a Variable




Selecting a user in XSOL Automation



Automation Systems Design only

You need to use a Business Rule if you need to design your model so that a user can be selected via a Worksheet in *XSOL Automation*.

To select a user in XSOL Automation:

- Click on the  Variables designer button in the **Design for Automation** window.
- Select the required **User** Variable from the Variable list, or create a new one (refer to page 128 for more information).
- Click on the  button to create a new Business Rule for the Variable. Rename the Business Rule, if required.
- Drag and drop any required Roles from the  Roles area in the right pane. In an *XSOL Automation* system you can select all Users who have one or more of these Roles assigned to them.

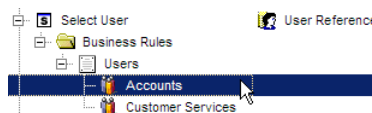


Figure 4.98: Creating a Variable and Business Rule that allows you to select a user in XSOL Automation

- You can now add this Variable to any required Worksheet. In a run-time system this will display a list of Users for you to select.

Changing a Variable



Automation Systems Design only

Changing a Variable involves adding or changing any properties for this item. Refer to page 128 for a detailed description of Variable properties.

To change a Variable:

1. Click on the Variables Used designer button in the **Design for Automation** window.
2. Click on the required Variable.
3. Change any properties in the **Properties Editor**, if required.

Property	Value
Caption	Is it a pricing issue ?
Data Type	Binary
Group	(unassigned)
Store variable	<input checked="" type="checkbox"/>
True Value	Yes
False Value	No
Default Value	<input checked="" type="checkbox"/>
Default Calculation	(None)
Mandatory	<input checked="" type="checkbox"/>

Figure 4.99: Changing a Variable

4.12.2 Using Reference Variables



Automation Systems Design only

A Reference Variable selects an instance of a Table. A Reference Variable is automatically created for a Table whenever a Table is created, and has the same name as the Table. Reference Variables contain Datafield Variables, which enables a datafield to be modifiable in a Worksheet. You can create more than one Reference Variable for a particular Table, if required. For example, you may need to select more than one product from the Product Table on a Worksheet.

To add a Reference Variable:

1. Click on the Ref Variables designer button in the **Design for Automation** window.
2. Click on the Reference Variable Group that you would like to add a Reference Variable to (refer to page 32 for more information if you want to add Group).
3. Click on Table that you want to create a Reference Variable for from the Reference Data list at the right side of the window.
4. Drag and drop the Table onto the specific Table type in the Reference Variables list in the middle of the window.
5. Enter any information for the Reference Variable in the **Properties Editor**, including User Notes, if required. Refer to page 145 if you do not know how to add a User Note.
6. Refer to page 135 if you need to create any Datafield Variables for the new Reference Variable.

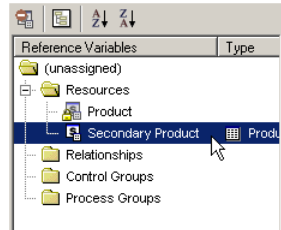


Figure 4.100: Adding a Reference Variable


Changing a Reference Variable



Automation Systems Design only

Changing a Reference Variable involves adding or changing any properties for this item. Refer to page 133 for a detailed description of Reference Variable properties.

To change a Reference Variable:

1. Click on the  designer button in the **Design for Automation** window.
2. Click on the required Reference Variable.
3. Change any properties in the **Properties Editor**, if required.



Using Business Rules with Reference Variables



Automation Systems Design only

Business Rules are used to restrict the information that a user can select. Business Rules are usually used on Reference Variables to find a particular instance of a Table and return the value, but they can also be used to restrict the users or documents that a user can select (refer to page 132 and page 42 for more information).

To add a Business Rule to a Reference Variable:

1. Click on the  designer button.
2. Click on the required Reference Variable in the left pane.
3. Click on the  button to create a new Business Rule for the Variable. Rename the Business Rule, if required.
4. Select the required datafield from the right column drop-down list for each listed UI field. This restricts the returned values by the selected fields.

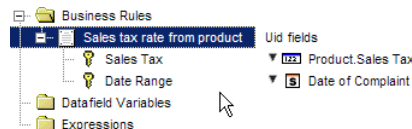


Figure 4.101: Adding a Business Rule to a Reference Variable

Using Datafield Variables



Automation Systems Design only

A Datafield Variable is used to allow a datafield's value to be changed within a Process. Datafield Variables have a **Change With Instance** property. This determines whether the run-time system refreshes the Datafield Variable or not when a field that uses the Datafield Variable is changed. The available options are:

- **Always** - Always refresh the Datafield Variable when creating a new instance.
- **Never** - Never refresh the Datafield Variable when creating a new instance.
- **Prompt** - Ask the user whether they want to refresh the Datafield Variable when creating a new instance.

To add a Datafield Variable:

1. Click on the designer button in the **Design for Automation** window.
2. Click on the Reference Variable to which you would like to add a Datafield Variable.
3. Click on one or more of the datafields displayed in the Reference Data list and drag them onto the Reference Variable. You can only drag data fields from the same Table that the Reference Variable points to.
4. Enter any information for the Reference Variable in the **Properties Editor**, including User Notes, if required. Refer to page 145 if you do not know how to add a User Note. Alter the **Change With Instance** setting, if required.

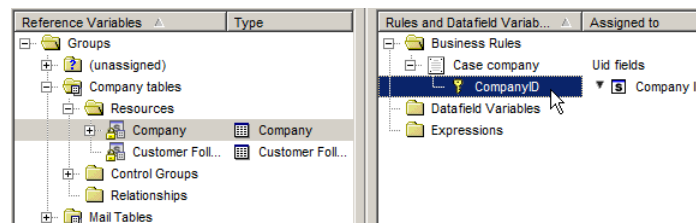


Figure 4.102: Adding a Datafield Variable

Changing a Datafield Variable



Automation Systems Design only

Changing a Datafield Variable involves adding or changing any properties for this item. Refer to page 135 for a detailed description of Datafield Variable properties.

To change a Datafield Variable:

1. Click on the designer button in the **Design for Automation** window.
2. Click on the required Datafield Variable.
3. Change any properties in the **Properties Editor**, if required.

4.13 Designing Lookup Lists



Automation Systems Design only

A Lookup List displays a selection of data from one or more Advanced Queries. Lookup Lists are used in Worksheets and allow you to present only the data that you want to a user at a particular point in a Process.



If you want a an *XSOL Automation* user to be able to run Lookup Lists from a Role folder, create a Role called "Lookups" and in the XSOL Security Administrator application assign it to the user.

Lookup List properties:

- **Default Execution Level** - Identifies whether the Roles assigned to the Lookup List can (**No one except assigned Roles**) or cannot (**Everyone except assigned Roles**) initiate the Lookup List.
- **Primary table** -Select the Table that provides the data to the Lookup List.
- **Maintenance task** - Select the Maintenance Task Worksheet that the Lookup List can open in *XSOL Automation*.

To add a Lookup List:

1. Click on the Enterprise or Trading Entity that you require in the Hierarchy list in the **Design Enterprise** window.
2. Click on the button.
3. Click on the Lookup Lists designer button.
4. Click on the Lookup List Group that you want to add this Lookup List to. Refer to page 32 for more information about using Groups.
5. Click on the button.
6. Enter the name of the Lookup List and press the **Enter** key to create another item or press the **Tab** key to only create this item.
7. Choose the required Advanced Query that you want to add datafields from in the **AdvancedQuery** drop-down list in the **Properties Editor**.
8. Click on the required Maintenance Task from the **Maintenance task** drop-down list in the **Properties Editor**, if you want users to run the associated Maintenance Task from the Lookup List.
9. Select one or more datafields from the selected Advanced Query and drag and drop them onto the new Lookup List.

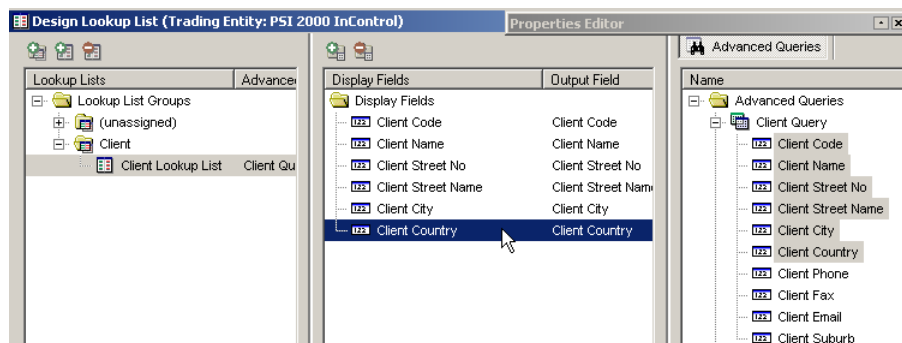



Figure 4.103: Adding a Lookup List

10. Repeat the last two steps for as many datafields as you require from one or more Advanced Queries.
11. Add or alter the properties of the Lookup List, if required.
12. If you want the Lookup List to be accessible by users in *XSOL Automation*, click on the  **Roles** button in the right pane.
13. Drag and drop any required Roles onto the Lookup List in the left pane.


4.13.1 Changing a Lookup List



Automation Systems Design only

Changing a Lookup List involves adding or changing any properties for this item. Refer to page 136 for a detailed description of Lookup List properties.

To change a Lookup List:

1. Click on the  **Lookup Lists** designer button in the **Design for Automation** window.
2. Click on the required Lookup List.
3. Change any properties in the **Properties Editor**, if required.
4. Add or remove datafields to the Lookup List, if required.

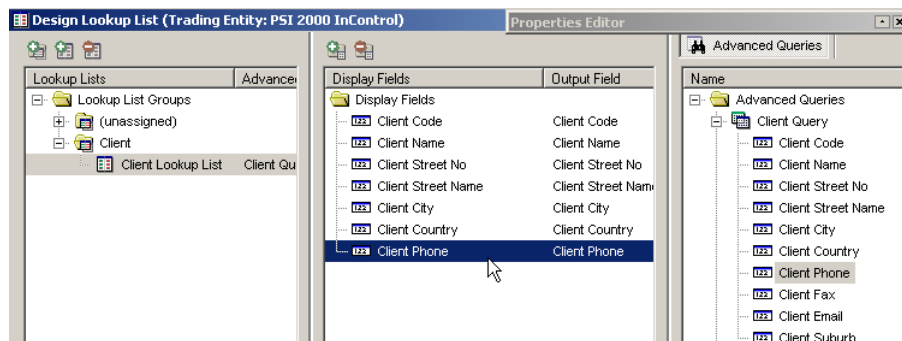


Figure 4.104: Changing a Lookup List

4.14 Designing Expressions



Automation Systems Design only

An Expression is a piece of business logic that can optionally include conditional statements the output of which is a result that is either stored in a Variable or queried by the Process Manager to determine the required future actions. For example, **Price * Quantity** represents the value of multiplying the Variables Price and Quantity.

XSOL Mapping also enables you to define Macro Expressions. A Macro Expression is an Expression or calculation, as defined above, which uses Parameters for the Variable elements (e.g. Quantity, Price) within the Expression or calculation. A Macro Expression is a define-once/use-many entity in that it can be assigned to many Process Variables. By using Macro Expressions any change applied to that Macro Expression are automatically applied to all Process Variables that the Macro Expression is assigned to.

Expressions can be defined in several parts of *XSOL Mapping*, including:




- **Variables** - Used to determine the content of the Variable. Displayed in the **Expression** property.
- **Condition and Decision nodes** - Used to activate the next Stage or Task in Process Flow. Displayed in the **Expression** property for Conditions or in the Decision Table for Decisions.
- **Stage scheduling** - Used to schedule when the next Stage is activated. Displayed in the **Schedule Expression** property and must return a date time value.
- **Stage summary** - Used to provide distinguishing information on the Stage that appears in the first column of the To Do List. Displayed in the **Stage Summary for To-do List** property.
- **Time Calculation** - Used to provide a time period before the next Escalation Level is initiated. Displayed in the **Timeout Calculation** property.
- **Macro Expressions** - Refer to the previous text for more information. Displayed in the **Expression** property.



The **Expression Editor** uses the Python scripting engine to function. Refer to <http://www.python.org/doc/> for in depth information about Python, if required.

Macro Expression properties:

- **Result Type** - Determines the data type used for the result of the Macro Expression (refer to page 119 for more information). Select the required data type from the drop-down list.

To add an Expression or Macro Expression:

1. Use the following instructions that apply to you.
 - (a) If you need to add an Expression to a Variable, click on the  Variables Used designer button in the **Design for Automation** window, click on the required Task in the left pane, click on the required Variable in the middle pane, and click on the  button.
 - (b) If you need to add an Expression to a Condition or Decision node, click on the required node in the **Design Process Flow** window and double-click on the **Expression** field in the **Properties Editor**.
 - (c) If you need to add an Expression to a Stage, click on the  button in the **Design for Mapping** window, click on the required Stage, and double-click on the **Schedule Expression** or **Stage Summary for To-do List** field in the **Properties Editor**.

- (d) If you need to add a Macro Expression, open the **Design Enterprise** window and click on the  button.
-  Do not use any of Python's reserved keywords as names, as they might cause an error in Python - refer to page 198 for more information.
- For Variable Expressions or Macro Expressions, enter the name of the Expression and press the **Enter** key to create another item or press the **Tab** key to only create this item.
 - If you have created a Macro Expression, select the data type that the Expression returns from the **Result Type** drop-down list in the **Properties Editor**. Refer to page 119 for more information about data types.
 - For Variable Expressions or Macro Expressions, double-click on the **Expression** field in the **Properties Editor**.
 - You are now in the **Expression Editor** window. If you are creating a Macro Expression, you will need to add one or more Parameters. Refer to page 141 for more information. You can add a Parameter to the script by double-clicking on the Parameter.
 - You now need to add some logic to the Expression. The **Expression Editor** uses the Python scripting engine to function. The following items in the Editor show you the various functions, operators, and data types that you can use - click on an item to insert it into the script.
 - **Functions** - The Functions list displays an extensive list of flow control and data types.

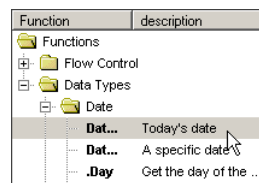



Figure 4.105: Functions list

- **Operators** - The Operators list displays an extensive list of scripting operators. You can also access various operators by clicking on the  button.

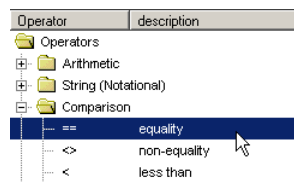



Figure 4.106: Operators list

Refer to <http://www.python.org/doc/> for in depth information about Python, if required.

- To add Variables to the Expression, click on the  Variables designer button in the right pane and drag and drop the required Variables to the main display area. You can use Variables that are not available in the list by using the External ID of the Variable (refer to the **Properties Editor** for this information).



You can use different buttons in the right pane to add Macro Expressions, Document Locations (page 41), or System Values (page 43).

- The last line of all Expressions must start with **return**. This returns the value of the Expression to the system.

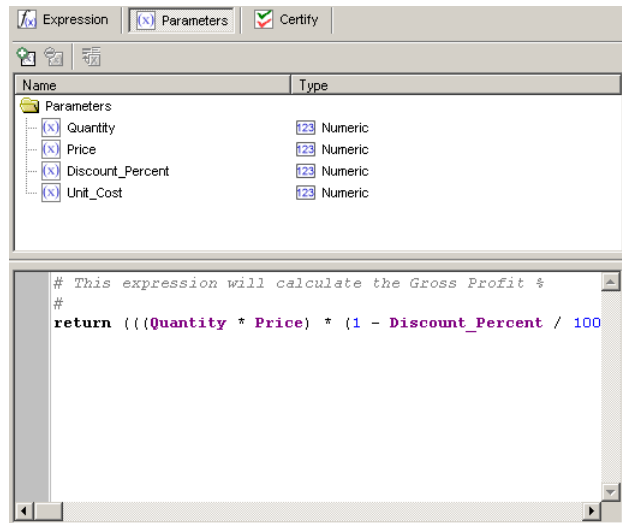


Figure 4.107: Adding a Macro Expression



4.14.1 Assigning an Expression



Automation Systems Design only

Expressions need to be assigned to the Variable that they are attached to in order to be active in the run-time system. Expressions are automatically assigned to the Variable when first created.

To assign or unassign an Expression:

- Click on the  Variables Used designer button.
- Click on the required Expression attached to a Variable.
- Click on the  button to assign or unassign the Expression.

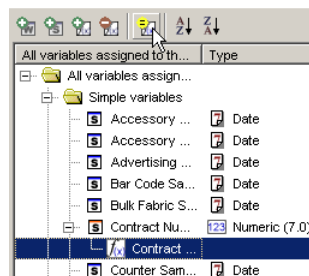


Figure 4.108: Assigning an Expression

4.14.2 Changing an Expression



Automation Systems Design only

Changing an Expression can involve altering the Expression's name, changing any items that you have added to the Expression's properties, and altering the logic of the Expression. Refer to page 138 for a detailed description of Expression properties.



If you are changing a Macro Expression, you may need to change its Parameters as well. Refer to page 142 for more information.

To change an Expression:

1. Click on the required Expression and change any properties of that Expression in the **Properties Editor**, if required.
2. Open the **Expression Editor** for the required Expression.
3. Change the script, if required. You may need to change the script's parameters if this is a Macro Expression.

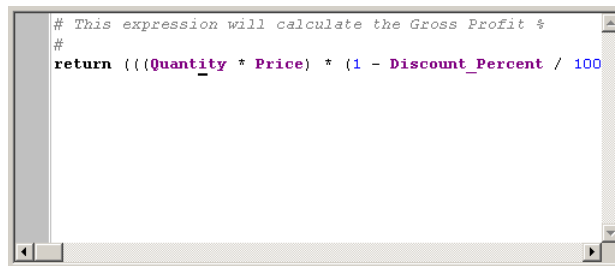


Figure 4.109: Changing an Expression



4.14.3 Using Parameters



Automation Systems Design only

Parameters are used in Macro Expressions and are simply generic names for Variables used within the Expression. The Parameters are substituted by actual Process Variable names when the Macro Expression is assigned to a Process Variable, which will hold the result of the Macro Expression when it is run. Parameters are also used for testing Macro Expressions, as you can substitute test values for the Parameters of a Macro Expression when you test it (refer to page 143 for more information).

To add a Parameter

1. Open the **Expression Editor** for the required Macro Expression.
2. Click on the  **Parameters** designer button.
3. Click on the  button.
4. Enter the name of the Parameter and press the **Enter** key.

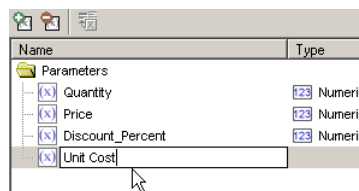


Figure 4.110: Adding a Parameter

5. Repeat the previous step if you want to create more Parameters. Otherwise press the **Esc** key to delete the unwanted Parameter.
6. Select a data type for each new Parameter from the **Data Type** field in the **Properties Editor**.

Changing a Parameter



Automation Systems Design only

Changing a Parameter involves adding or changing any properties for this item.

To change a Parameter:

1. Open the **Expression Editor** for the required Macro Expression.
2. Change any properties in the **Properties Editor**, if required.

4.14.4 Checking an Expression's syntax



Automation Systems Design only

You need to check an Expression's syntax once you have finished writing the logic of the Expression. This makes sure that the Expression will run.



The **Expression Editor** uses the Python scripting engine to function. If required, refer to page 197 for more information about the custom changes made to Python in *XSOL Mapping*, or refer to <http://www.python.org/doc/> for in-depth information about standard Python.

To check an Expression's syntax:

1. Open the **Expression Editor** for the required Expression or Macro Expression.
2. Click on the **Check Syntax** button. Check the Messages area for the results of the check.

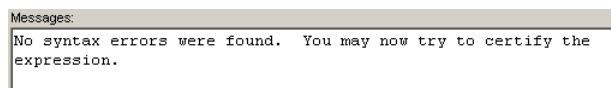


Figure 4.111: No errors found

Any errors are highlighted in the Expression's script.

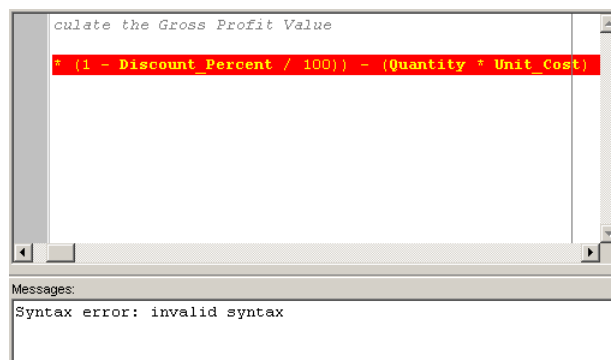


Figure 4.112: Invalid syntax

4.14.5 Certifying an Expression



Automation Systems Design only

Certifying an Expression enables you to run an Expression in a test environment. This enables you to determine whether the Expression will run as you expect, including sample Parameter values and an expected result. You can create more than one certification test, if required, and run all of them simultaneously.



You need to check the syntax of the Expression before you can certify it (refer to page 142 for more information).

To certify an expression:

1. Check the Expression's syntax (refer to page 142 for more information). You always need to do this, unless there have been no changes made to the Expression's script.
2. Click on the **Certify** button.
3. Click on the button to create a new test, if required.



You can delete a certification test by clicking on it and clicking on the button.

4. Click on the required certification test and click on the button, or press **F5**.

To run all certification tests for this expression simultaneously, click on the button.

5. Click on the **Yes** button if it appears.
6. You may need to put in sample values for each Parameter. Click on the **Value** column next to the first Parameter and enter a value.



If you do not want to include the expected result as part of the certification test, de-select the **Expected Result** tick box.

7. Press the **Tab** key and repeat the last step and this step for each Parameter in the list.

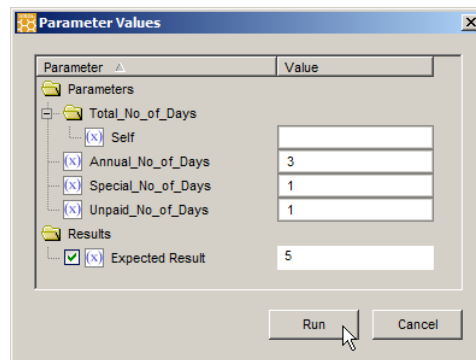


Figure 4.113: Certifying an Expression

8. Click on the **Run** button. Check the **Messages** area for the result of running the Expression.

4.14.6 Certifying all Expressions




Automation Systems Design only

You can certify all Expressions in a model in one action. This enables you to determine whether all Expressions will run as you expect.



Any Parameter values need to be entered into specific certification tests before you can certify all Expressions.

To certify all Expressions:

1. Click on the  button in the Tool bar.
2. Click on the **Close** button when you have finished reviewing the certification test results.

4.15 Using User Notes

User Notes enable you to add extra information relevant to an item. Any item in *XSOL Mapping* can have one or more User Notes added to it, and you can specify whether a User Note is specific to a particular item or generally available for all items. Each User Note has a defined type, such as Numeric or Word, which allows you to enter specific types of information. You can define any number of User Note fields with their own captions, and you can specify which of them will be included in the generated documentation. User Notes are a useful way of adding important information to the model.






There are two permanent User Note fields besides the user-defined User Note fields:

- **Version** - Used for entering version control information. Version information is included in the generated documentation.
- **Designer Comments** - Used for entering information from the model designer on how the model was designed. Designer Comments are not included in the generated documentation.



You need to define one or more User Notes before you can use User Notes (refer to page 45 for more information).

To add a User Note:

1. Click on the required item and open the **Properties Editor**, if it is not already displayed.
2. Click on the required User Note field (displayed underneath the **User Defined Notes** heading).
3. Enter information into the User Note in the following manner:
 - If you are using a **Binary** User Note, select the required true, false, or unknown value from the drop-down list.
 - If you are using a **Date** User Note, click on the  button and select the required date. You can also manually enter the date in the format **dd/mm/yy**.
 - If you are using a **Datetime** User Note, click on the  button and select the required date. You will then need to enter the required time in a 24-hour format, each time unit separated by colons. For example, for 8:45pm enter **20:45:00**.
 - If you are using a **Word** User Note, click on the  button and enter the User Note text in the displayed window. You can change the appearance of the text using all the controls available in the User Notes Editor window, including changing font size, adding bullet points, indenting text, etc. Refer to page 146 for more information about entering information in this User Note type.
 - If you are using a **Memo** User Note, click on the  button and enter the User Note text in the displayed window. You can enter multiple lines of text, but you can not format the text (change font, add italics, etc.).
 - If you are using a **Notational** User Note, enter one line of text in the field.
 - If you are using a **Numeric** User Note, enter the required numbers in the field.
 - If you are using a **Percent** User Note, enter the required percentage numbers in the field.
 - If you are using a **Scientific** User Note, enter the required numbers in the field.
 - If you are using a **Time** User Note, enter the time in a 24-hour format, each time unit separated by colons. For example, for 8:45pm enter **20:45:00**.
4. The item will now have an icon next to it -  - to indicate that one or more User Notes have been defined for this item.

4.15.1 Using Word User Notes

The following Microsoft Word functionality is not supported when using Word-type User Notes:

- Shading
- Page Borders
- Backgrounds
- Table of contents / Table of figures / Index
- Headers / Footers / Endnotes
- Centre / Right / Justify alignment
- Clip Art
- Objects (using the **Insert >Object...** menu option). This includes all objects, such as bitmaps, calendars, media clips, etc.

Text boxes work correctly as long as the textbox is not the last item in the User Notes Editor.

Please read the following text if you need to insert an image or link, or need to copy information from other Office applications.

1. If you need to insert an image, click on the **Insert >Picture >From File...** menu item, select the required image, and click on the **Insert >Link to File** menu option to select the required image file.
2. If you need to insert a link to a file or website, select the required text that will become the link and click on the **Insert >Hyperlink...** menu item. Fill in the required details and click on the **OK** button. You can insert links to websites, FTP (File Transfer Protocol) sites, local files, as well as links that open the user's email application.

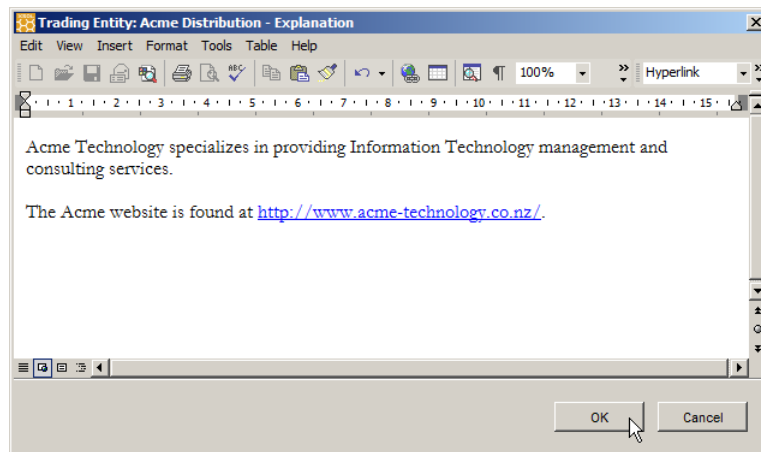


Figure 4.114: Adding a Designer User Note

3. If you need to copy information from another Office application, such as Excel or PowerPoint (not including Word), you need to click on the **Edit >Paste Special...** menu item. Double-click on **HTML Format** if the copied information is formatted text, or **Picture (Enhanced Metafile)** if the copied information is a graphic.

4.15.2 Using the "Normal" style with User Notes

The Word-type User Notes Editor is a true WYSIWYG (What You See Is What You Get) editor. A WYSIWYG application is one that enables you to see on the display screen exactly what will appear when the output is created. This is achieved through the User Notes Editor using the "Normal" style in the **Normal.dot** file used by your installation of Microsoft Word. All non-styled content (text that had not been altered by changing font size or type) displays as per the "Normal" style.



Each *XSOL Mapping* user will see un-styled content using their "Normal" style, and will be saved accordingly if the User Note is changed. Multiple-user environments will need to co-ordinate their versions of **Normal.dot** to avoid confusion. As soon as any content in a User Note is styled (the font size or type are changed), the User Notes Editor stops using the "Normal" style as a reference. This content will appear exactly the same for all *XSOL Mapping* users, and requires no co-ordination in multiple user environments with the "Normal" style and **Normal.dot**.

If you want new User Notes to always appear with the same default style settings (such as 12 point Arial) you need to change the "Normal" style in your version of **Normal.dot**. Please see your network administrator as to whether this is possible.

4.15.3 Changing a User Note

Changing a User Note involving altering the contents of the User Note.

To change a User Note:

1. Double-click on the required User Note field (displayed underneath the **User Defined Notes** heading) for the required item.
2. Change the User Note text in the displayed window. You can change the appearance of the text using all the controls available in the User Notes Editor window, including changing font size, adding bullet points, indenting text, etc.
3. Click on the **OK** button when you have finished changing the User Note.

4.16 Validation



Automation Systems Design only

Validation ensures that your model is logically correct. For example, a Process must have one Go and one or more Stop nodes. You can save the model without doing this, but the model is not valid. Any items in the model that are invalid by default are displayed as red (you can change this setting - refer to page 171 for more information). You can quickly see the reason why a red item is invalid by placing the mouse over the item - an explanation is displayed in pop-up text.

There are two ways that you can formally validate the model:

- Entire model
- One section of the model (for example, Tables)

It is a good idea to occasionally validate the section you are working on, and the entire model. However, if you are not creating a run-time system, validation is not required.



Validating Process Flow diagrams is not covered in this section, please refer to page 95 for more information about validating Process Flow diagrams.

4.16.1 Validating the model



Automation Systems Design only

A model must be validate before it can be committed to *XSOL Automation*. You can valid the entire model at any time.

To validate the model:

1. Click on the button in the top toolbar, underneath the pull-down menus.
2. Any items that do not validate are listed in red with their location in the model. If there are no red items, the model is valid, and you do not need to continue with this procedure.
3. Click on any red item. A description of why that item is invalid appears in the **Reasons** button.

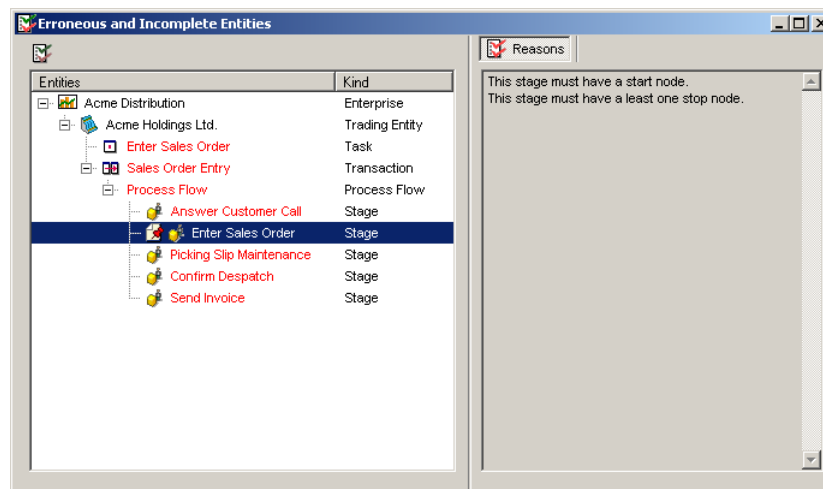



Figure 4.115: Validating the model

4. Some errors you will be able to fix using the **Properties Editor**. Click on the **Properties Editor** and alter the item's properties to make it valid.
Other errors will need to be fixed in the Process Flow diagrams (refer to page 80 for more information).
5. Repeat the last two steps for all red items.
6. Click on the  button. Repeat the procedure until you do not get any invalid items.



4.16.2 Validating specific items



Automation Systems Design only

You can validate a specific item or section at any time.

To validate a specific item:

1. Click on the item that you want to validate.
2. Click on the  button above the required item.
3. Any items that do not validate are listed in red with their location in the model. If there are no red items, the model is valid, and you do not need to continue with this procedure.
4. Click on any red item. A description of why that item is invalid appears in the  **Reasons** button.

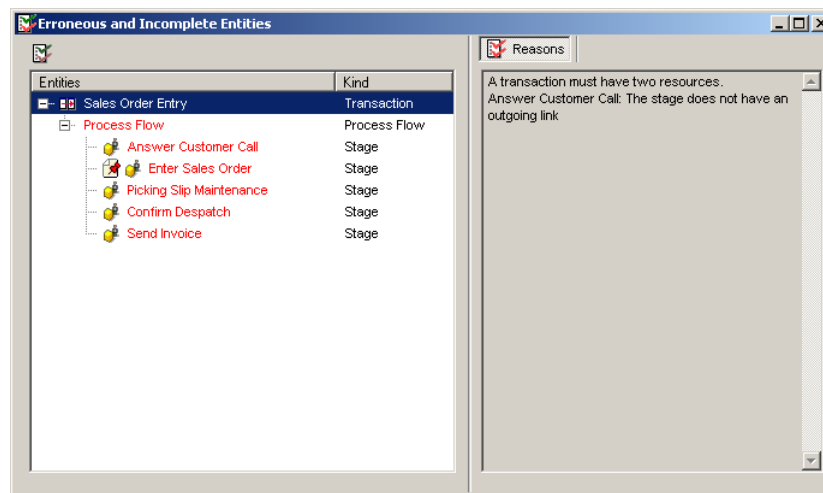




Figure 4.116: Validating the model

5. Some errors you will be able to fix using the **Properties Editor**. Click on the **Properties Editor** and alter the item's properties to make it valid.
Other errors will need to be fixed in the Process Flow diagrams (refer to page 80 for more information).
6. Repeat the last two steps for all red items.
7. Click on the  button. Repeat the procedure until you do not get any invalid items.

4.17 Where Used

The Where Used function allows you to check where any particular item is used in the model. This is useful for tracking purposes and is required if you need to delete an item that is used throughout the model.

To check where an item is used:

1. Click on the item that you want to check.
2. Click on the  button.
3. The Where Used window illustrates where the item is used in the model. If you need to change any of these links you need to do so manually.

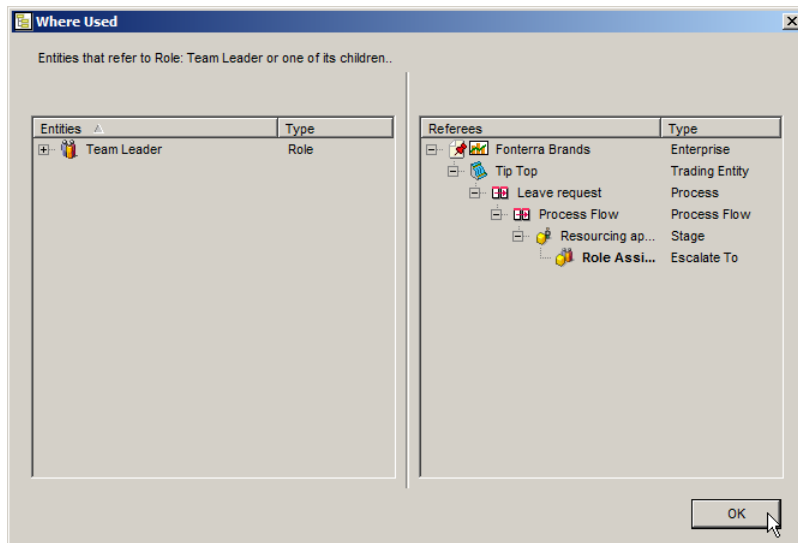


Figure 4.117: Checking where an item is used

4. Click on the **OK** button when you have finished using the window.

4.18 Generating documentation

XSOL Mapping allows you to generate different documents based on the model you are currently using. This is a completely automated procedure - all the text, page formatting, table of contents, index, and graphics are generated automatically from the model. The following formats are available:

- **Web sites** - To view web site documentation you need Internet Explorer 6, Service Pack 1 or higher.



All HTML outputs include the same model information, they are just displayed differently. Try generating all outputs and decide which one you prefer.

- **One page web site** - Generates a 1 page web site.
- **Multiple page web site** - Generates a many page web site.
- **HTML Help compiled help file** - Generates a compressed HTML Help file (.chm). This Microsoft format includes a searching mechanism and allows you to bookmark pages that you refer to often. You need to have the HTML Help Workshop installed on your system to generate this documentation (refer to page 161 for more information).

- **Microsoft Word** - To view or print the Word documentation you need Word 2000 or later installed on your system.



There may be downward-compatibility issues with Word documents generated using a newer version of Word and viewed using a previous version of Word. Please test your generated documentation if you are sending the Word output to someone who is using a previous version of Word.

- **Word document** - Generates a portrait-orientation Word document. All information is displayed in lists.
- **Tabular Word document (processes only)** - Generates a landscape-orientation Word document. Most information is displayed in a tabular format. To create diagrams that print properly for this option, you need to choose the **Custom Size** option for each diagram, and set the **Custom Width** and **Custom Height** attributes as the reverse of the paper size you are using. As an example, for A4 paper you need to set **Custom Width** as 297 and **Custom Height** as 210. Refer to page 167 for more information about making these changes.
- **Notes Report Word document** - Generates a portrait-orientated Word document. Only items with attached User Notes is displayed. You can control what User Notes are displayed by selecting the **Print in documentation** property at the model level - refer to page 45 for more information. If an entity does not have any printable User Notes attached to it, it will not be printed.


- **Spreadsheet** - To view or print spreadsheet documentation you need a spreadsheet application that can read a tab-delimited text file (such as Microsoft Excel).

- **Model hierarchy Excel document** - Generates a tab-delimited text file that you can open in your favorite spreadsheet application. All User Note types (except for Word and Memo) are supported, as well as Reference information (refer to page 145 and/or page 100 for more information). Once in that application, you can save it as a native file (e.g. .xls) and make any other changes you want.

4.18.1 Generating documentation outputs

Using *XSOL Mapping* you can generate documentation that describes your model. You can select what areas of the model you want to include in the generated documentation, and all generated files include a table of contents, a list of figures used, and an index.

To generate documentation:

1. Click on the  button.
2. Click on the documentation filter you want to use from the **Documentation Filters** area at the top of the window. Refer to page 152 if you need to add, edit, or delete a filter.
3. Click on the documentation layout you want to use from the **Documentation Layouts** area at the bottom of the window. Refer to page 154 if you need to add, edit, or delete a layout.
4. Click on the **Generate** button.

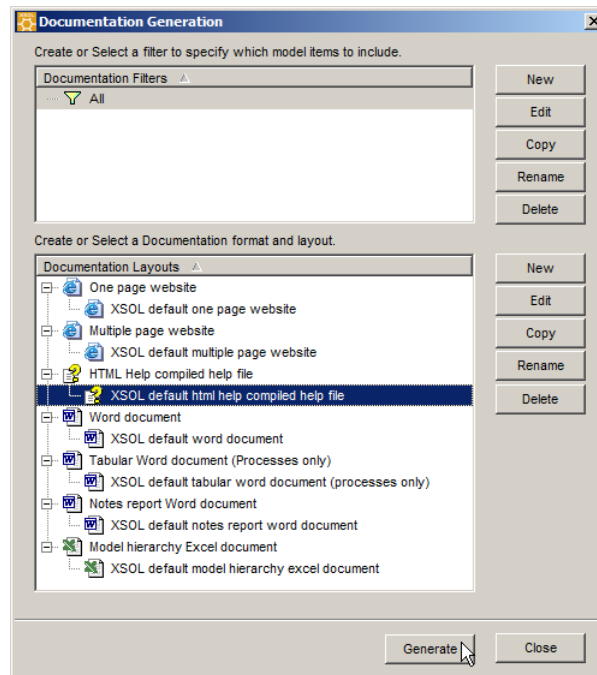


Figure 4.118:

5. Browse to the location where you want to save the document. Enter the name of the document in the **File name** field, and click on the **Save** button.
6. The documentation is now being generated. When the document is finished you can click on the **OK** button to view the document.


If you need to email the generated HTML output, please refer to page 161 for more information.

4.18.2 Using documentation filters

Documentation filters allow you to select the specific information from a model (such as Processes or Roles) that you want to include in the generated documentation. Filters are different from layouts - filters select the content to be included in the documentation, whereas layouts select the sections in the documentation that the content is put into (refer to page 154 for more information). All filters are model-specific and are stored directly in the model.

To add, edit, copy, rename, or delete a documentation filter:

1. Follow one or more of the following instructions:
 - To **add** a documentation filter, click on the **Add** button. Enter the name for the new filter and press the **Enter** key. Click on the **Edit** button and go to the next step in this procedure to edit the filter.

- To **edit** a documentation filter, click on the filter that you want to edit and click on the **Edit** button. Go to the next step in this procedure to edit the filter.
 - To **copy** a documentation filter, click on the filter that you want to copy and click on the **Copy** button. Enter the name for the new filter and press the **Enter** key.
 - To **rename** a documentation filter, click on the required filter and click on the **Rename** button. Enter the new name for the filter and press the **Enter** key.
 - To **delete** a documentation filter, click on the required filter and click on the **Delete** button.
2. Select the types of items that you want to include in the filter, then click on the **Next** or **Finish** button.
-  Refer to page 153 for more information about how selecting or de-selecting Roles influences your generated documentation.
3. Select the specific items that you want to include in the filter from the Enterprise and/or each Trading Entity. You can turn on or off all items within the Enterprise, a Trading Entity, or folder by clicking on the tick box next to the item's name.

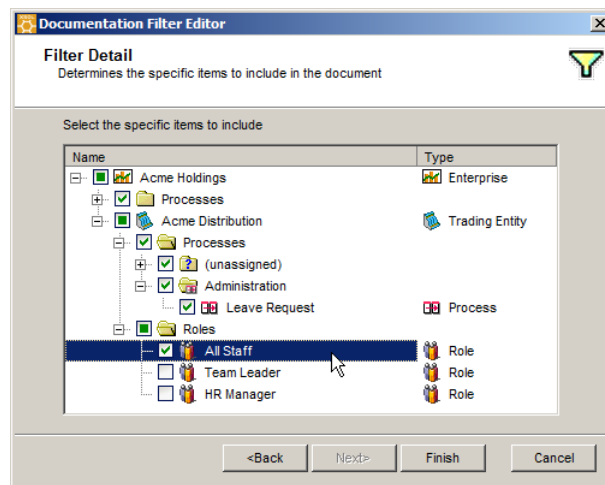


Figure 4.119: Editing a documentation filter

4. Click on the **Finish** button.

How selecting Roles changes your generated documentation

XSOL Mapping includes a feature that allows you to produce documentation for a specific Role, answering the question of: What exactly does this particular Role do? Obviously selecting only one Role restricts what content is included in the documentation, but you can also select any number of Roles. This topic explains how this restriction in content works.

Definitions:

- Role-related information - Any generated information that can include a reference to Roles at some point (Processes, Companion Processes, Process Summaries, Roles section, Maintenance Task Worksheets, Lookup Lists).
- Non-Role-related information - Any generated information that can not include a reference to Roles (Tables, Variables, front and back matter - table of contents, table of figures, etc.).

How it works:

- All Roles selected - Include both Role-related information and non-Role-related information. No restrictions that are caused by Roles apply.
- Some Roles selected - Include Role-related information (but only those that include the selected Roles) and non-Role-related information. This will remove all Role-related information that does not include that Role or Roles (including Processes with no Roles assigned).
- No Roles selected - Only include non-Role-related information. This will remove **all** Role-related information sections, (i.e. all Processes and Roles sections, including Processes with no Roles assigned).

4.18.3 Using documentation layouts

Documentation layouts allow you to specify the document sections that you want to include in the generated documentation. These sections can be turned on and off and re-ordered to suit your needs. You can also add custom sections for specific information that does not come from the model - refer to page 157 and/or page 162 for more information.


Layouts are different from filters - layouts select the sections in the documentation, whereas filters select the content to be included in those sections (refer to page 152 for more information).

Sections

- **Title page** - Generates a title page in Word documents, or a title heading and author information in HTML.
- **Table of contents** - Generates a table of contents.
- **Table of figures** - Generates a table of figures, which lists all generated diagrams in the documentation.
- **Preface** - Includes a preface section.
- **User Notes on User Notes** - Includes a section that includes the User Notes that have been attached to the User Note definitions, if any.
- **Multiple Trading Entities section** - Generates a section on the multiple trading entities in the model, if there are more than one.
- **Model and Enterprise User Notes** - Includes all model and Enterprise User Notes in a separate section, if there are any.
- **Enterprise and Trading Entity layout** - Determines the layout for the Enterprise and Trading Entity sections of the model.
- **Processes** - Includes all Process information in a separate section.
- **User Notes** - Includes all User Notes for Processes.
- **Process Flow diagrams** - Includes all Stage Process Flow diagrams in each Process.
- **Who can initiate?** - Includes information on who can start each Process.
- **Companion Processes** - Includes all Companion Processes in each Process or Stage (depending on where in the section hierarchy this occurs).
- **Process Summaries** - Includes all Process Summaries in each Process.
- **Stages Roles Tasks** - Includes all Stages in each Process.
- **Stage User Notes** - Includes all User Notes for Stages.

- **Stage Flow diagrams** - Includes all Task Process Flow diagrams in each Stage. The **Stage** option must be selected to include this output.
- **Roles and Escalation Levels** - Includes all Roles and Escalation Levels (everything in the **Roles and Escalations** folder) in each Process - refer to page 102 for more information.
- **Tasks** - Includes all Tasks in each Stage. The **Stage** option must be selected to include this output.
- **Roles section** - Includes a detailed section on each Role in the model.
- **Table Groups** - Includes all Table Groups and Tables in a separate section.
- **Resources** - Includes all Resources in each Table Group in a separate section.
- **Resource Extensions** - Includes all Resource Extensions in each Table Group in a separate section.
- **Control Groups** - Includes all Control Groups in each Table Group in a separate section.
- **Relationships** - Includes all Relationships in each Table Group in a separate section.
- **Maintenance Task Worksheets** - Includes all Maintenance Task Worksheets in a separate section.
- **Process Variables** - Includes all Process Variables in a separate section.
- **Task Worksheets** - Includes all Task Worksheets in an appendix.
- **Index** - Generates an index.

Options

- **User Notes** - Includes all User Notes. User Notes can be individually included or excluded in the output in the  designer button - refer to page 45 for more information.
- **Conditions** - Includes all Conditions nodes in each process.
- **Decisions** - Includes all Decisions nodes in each process.
- **Document Information** - Includes when the document was generated and when the model was last modified. The **Title page** section must be selected to include this output.
- **Task Types** - Includes Task Type details after each Task. The **Tasks** section must be selected to include this output.
- **Use bullets instead of numbers for Stage and Task lists** - Stages and Tasks are displayed as bullets. If not selected, Stages and Tasks are displayed as numbers.
- **Worksheets as images** - Task Worksheets are generated as images and appear in both the HTML and Word outputs. If not selected, Task Worksheets are generated as HTML forms. These appear in the HTML outputs, but not Word.



The generated HTML forms approximate what you see in the **Design Worksheet Layout** window. The layout is accurate, but the visual appearance will be different from *XSOL Mapping*.

- **Include Escalation Levels in selected Roles output** - Includes Roles in Escalation Levels when generating documentation focused on one or more Roles - refer to page 151 for more information. If not selected, only Roles that are not in Escalation Levels are included.



This option is ignored if all Roles for a Trading Entity or Enterprise are selected.

- **Diagram Captions** - Adds a caption to all generated diagrams. If not selected, the table of figures will contain a warning.

- **Decision Tables** - Includes Decision outcomes in a separate table. The **Decisions** section must be selected to include this output.
- **Datafield Subheadings** - Groups all datafields in Tables into separate groups (such as "Fields in UID" or "Auto Display Fields"). If not selected, all datafields are listed alphabetically per Table.
- **Datafield/Variable Details** - Includes detailed information about any displayed Datafield or Variable in any section. If not selected, only the Datafield or Variable caption is displayed.
- **Business Rules** - Includes Business Rule information for Tasks. The **Stages Roles Tasks** option must be selected to include this output.

To add, edit, copy, rename, or delete a documentation layout:

1.
 - To **add** a documentation layout, click on the **Add** button. Enter the name for the new layout and press the **Enter** key. Click on the **Edit** button and go to the next step in this procedure to edit the layout.
 - To **edit** a documentation layout, click on the layout that you want to edit and click on the **Edit** button. Go to the next step in this procedure to edit the layout.
 - To **copy** a documentation layout, click on the layout that you want to copy and click on the **Copy** button. Enter the name for the new layout and press the **Enter** key.
 - To **rename** a documentation layout, click on the required layout and click on the **Rename** button. Enter the new name for the layout and press the **Enter** key.
 - To **delete** a documentation layout, click on the required layout and click on the **Delete** button.
2. If you selected a Microsoft Word output, select the Word version that you want to use to edit this layout and click on the **Next** or **Finish** button. If you select a version of Word that is not installed on your machine, you will still be able to edit the layout but you will not be able to generate documentation using this layout.
3. Select the template for the generated document from the **HTML template** or **Word template** drop-down list. Refer to page 159 or page 164 for more information.
4. Select the sections that you want to include in the output from the left side of the window, as described earlier in this topic. You can insert custom sections from the right-side of the window - refer to page 157 or page 162 for more information.

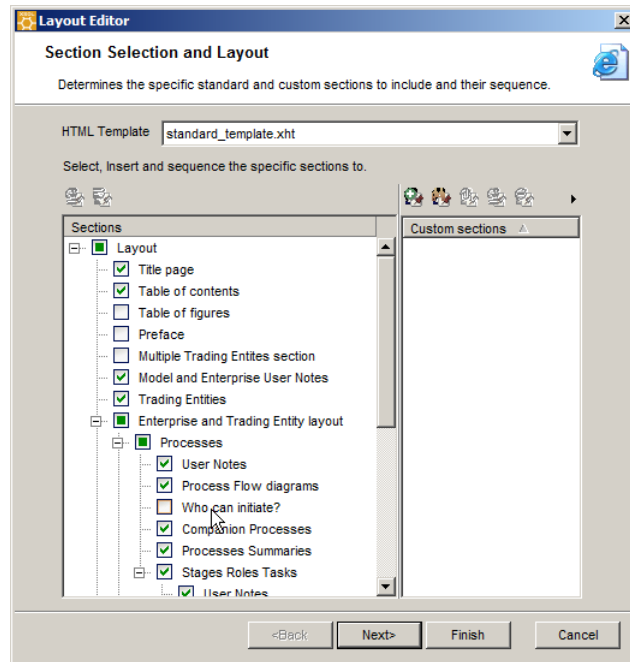


Figure 4.120: Editing a documentation layout

5. Click on the **Next** or **Finish**.
6. Select the options that you want to include in the output, as described earlier in this topic.
7. Click on the **Finish** button.

4.18.4 Using HTML custom sections

There are two ways to create HTML (HyperText Markup Language) custom sections, either through the in-built **Custom Section Editor** (recommended), or by creating your own HTML file. These sections can contain any amount of information and can be inserted one or more times into any place in the document layout. Custom sections are stored as files in the **Document template folder** with the extension **.xcf** (refer to page 175 for more information).



Custom sections that are inserted into a document layout in some locations will be repeated in the generated documentation. For example, inserting a custom section at the start of the **Processes** section will be repeated for every printed process.

To insert, create, add, edit, rename, or remove an HTML custom section:

1. Follow one or more of the following instructions:
 - (a) To **insert** a custom section, drag and drop the section from the **Custom sections** area on the right into the **Sections** area on the left.

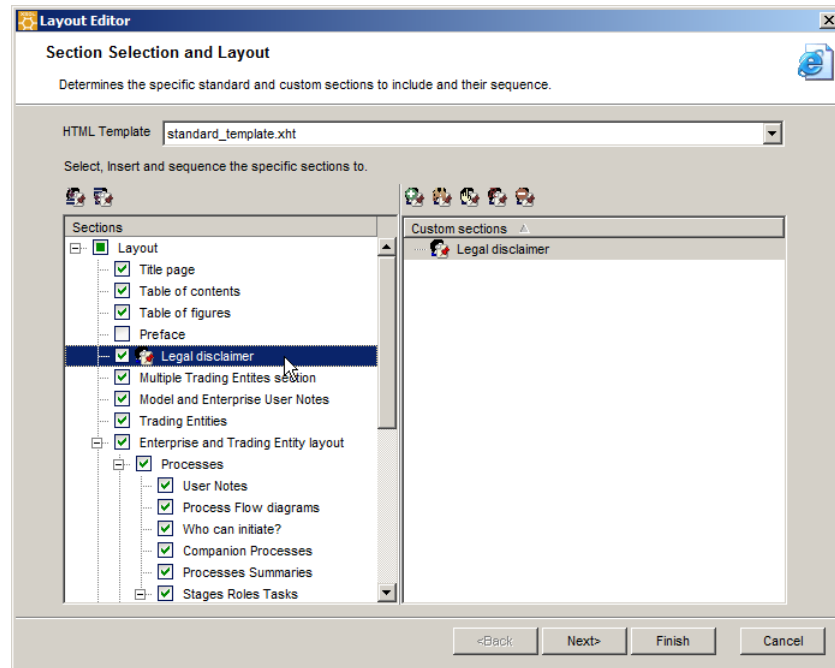









Figure 4.121: Inserting an HTML custom section

- (b) To **create** a custom section using the **Custom Section Editor**, click on the  button above the right area and rename the item. Click on the  button and go to the next step in this procedure to edit the section.

To **create** a custom section without using the **Custom Section Editor**, you can create an HTML page using any HTML editor. The following restrictions apply:

- The file must be valid XHTML (Extensible Hypertext Markup Language). For example, all tags must be well-formed, all element and attribute names must be in lower case, etc. Refer to <http://www.w3.org/TR/xhtml1/#diffs> for more information.
 - The file must be saved in the **Document template folder** directory with the extension **.html** or **.htm** (refer to page 175 for more information).
 - Only the contents of the **>** tag will be included in the generated documentation. All other code will be ignored.
- (c) To **add** a custom section from the list of custom sections already created in the **Document template folder** location, click on the  button above the right area, select the sections you are interested in using in this layout, and click on the **OK** button.
- (d) To **edit** a custom section, click on the required custom section in the right area and click on the  button above this area and go to the next step in this procedure to edit the section.
- (e) To **rename** a custom section, click on the required custom section in the right area and click on the  button above this area. Enter the new name for the section and press the **Enter** key.
- (f) To **remove** a custom section from this layout completely (including all references in the **Sections** area), click on the required custom section in the right area and click on the  button above this area. Click on the **Yes** button in the confirmation dialog.

To **remove** a custom section reference from this layout, click on the required custom section in the left area and click on the  button above this area.

2. You can change the appearance of the text using all the controls available in the User Notes Editor window, including changing font size, adding bullet points, indenting text, etc. Refer to page 146 for more information about entering information in this User Note type.

4.18.5 Using HTML documentation templates

All HTML (HyperText Markup Language) documentation is generated using a template. The template defines the HTML code that goes around the main content that comes from the *XSOL Mapping* model. You can edit the template to include content that goes before or after the main content (the web equivalent of a header or footer), add in extra style (CSS (Cascading Style Sheets)) information, add in Javascript code, etc.

You need the following in order to create these links:

1. Some knowledge of HTML. The amount of knowledge required varies depending on what you are trying to achieve.
2. An HTML editor. If you feel comfortable with editing HTML by hand, this can be as simple as a plain text editor (such as Notepad).

The following XSOL-specific tags are looked for by the documentation generation process:

- `<link d:id="XSOL_STYLESHEET" />` - Inserts a reference to the *XSOL Mapping* default CSS file, **generic.css**.
- `<title d:id="XSOL_TITLE" />` - Inserts a dynamically-generated title for the HTML page.
- `<p d:id="XSOL_MAIN_CONTENT" />` - Inserts the main content that comes from the model.



Removing this tag will remove almost all model-related information. Do not remove unless you know what you are doing.

The following restrictions apply:

- The file must be valid XHTML (Extensible Hypertext Markup Language). For example, all tags must be well-formed, all element and attribute names must be in lower case, etc. Refer to <http://www.w3.org/TR/xhtml1/#diffs> for more information.
- The file must be saved in the **Document template folder** directory with the extension **.xcf** (refer to page 175 for more information).
- The XSOL documentation namespace `xmlns:d="http://www.xsol.com/documentation"` must be declared in the `>` tag.

To use an HTML template:

1. Copy the default template **standard.template.xcf** in the **Document template folder** to the same directory (refer to page 175 for more information). Rename the template to a meaningful title.
2. Open the new template with your HTML editor.
3. Add, edit, or delete the HTML code, as required (the various XSOL-specific tags are described earlier in the topic). As long as the finished template is valid XHTML and includes the XSOL documentation namespace declaration, you can make any changes you want.



Javascript code that dynamically updates links to external files is not supported.

4. If you are linking to an external file, such as a logo image in the header or footer area (before or after the `<p d:id="XSOL_MAIN_CONTENT" />` tag), you need to do the following:
 - (a) Create a new directory in the same directory that the template is stored in with the name **TemplateName_files**. For example, if your template is called **acme.xcf**, the new directory must be called **acme_files**.
 - (b) Make all links in the template relative to the new directory. Absolute paths are only accidentally supported and will break in most situations.
 - (c) Copy all linked files into the new directory. These files will be copied into the generated HTML documentation automatically to ensure that your template works correctly.

4.18.6 Using styles with the HTML output

Most tags in the generated HTML (HyperText Markup Language) output have specific styles assigned to it using an external CSS (Cascading Style Sheets) file, which you can edit using standard CSS rules and syntax to conform to your requirements. This allows you to change the formatting of the HTML output very quickly.

You need the following in order to edit the CSS file:

1. Some knowledge of CSS and how it relates to HTML.
2. A CSS editor (often CSS functionality is included as part of an HTML editor). If you feel comfortable with editing CSS by hand, this can be as simple as a plain text editor (such as Notepad).

Any changes to the CSS file will automatically be applied as long as the CSS file has the correct name and is in the correct location.



Changing the style information in **generic.css** does **not** change User Note styles. All User Note style information is stored either in the model or in **Normal.dot** - refer to page 147 for more information.

To change the CSS file:

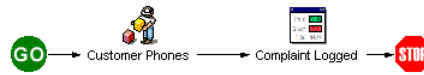
1. Make a backup of the **generic.css** file in the **Document template folder** (refer to page 175 for more information).
2. Open the **generic.css** file with your CSS editor.
3. Generate one of the HTML documentation outputs for the required *XSOL Mapping* model.
4. View the source code for the HTML page and find the tag and class attribute for the text that you want to change.
5. Change the CSS definition in the CSS file as required. For example, to make all paragraph tags Garamond instead of Arial, change the CSS entry to the following:

```
p {
  font-family: Garamond, serif;
  margin-top: 4px;
  margin-bottom: 2px;
}
```

6. Generate the HTML documentation again and check the results. Repeat the above steps as many times as required.

Stages, Roles, and Tasks

• Complaint Received



Action: Customer Complaints, Stage: Complaint Received

Roles:

Role assignments determine access to this Stage.

- Customer Services

Tasks:

- Customer Phones (Manual)
- Complaint Logged (Worksheet)

Figure 4.122: Garamond applied to all paragraph tags

7. Once you are happy with the output, make sure you take a backup of your custom CSS file, as the **generic.css** file will be replaced when you upgrade to a new version of the software.

4.18.7 Setting up HTML Help generation

You need to install Microsoft's HTML Help Workshop in order to produce the **HTML Help compiled help file** documentation (refer to page 151 for more information).

To set up HTML Help generation:

1. Download the HTML Help Workshop installer from Microsoft's web site - <http://go.microsoft.com/fwlink/?LinkId=14188>
2. Double-click on the installer file and follow all on-screen instructions. Although the default install location is recommended, you can install the application at any location on your machine.
3. Copy **hhc.exe** from the HTML Help Workshop installation folder to **C:\Program Files\XSOL\Mapping** (the XSOL installation directory).

You are now ready to produce **HTML Help compiled help file** documentation.

4.18.8 Emailing generated HTML documentation

In order to email the generated HTML documentation, both you and your email recipient(s) need to have an archiving tool available, such as 7-Zip or WinZip.

1. If you generated the **One page website** or **Multiple page website** documentation:
 - (a) Create a new ZIP archive which includes the created HTML file and a directory called "name_files". For example, if you generated an HTML file called "Acme Procedures", you need to archive both the "Acme Procedures.htm" file and the "Acme Procedures_files" directory. The "name_files" directory exists in the same directory as the generated HTML file.

If you generated the **HTML Help compiled help file** documentation:

- (a) Create a new ZIP archive which includes the created CHM file. For example, if you generated a CHM file called "Acme Procedures", you need to archive the "Acme Procedures.chm" file.
2. Attach the ZIP file to your email.

3. The recipients of the email need to save the email to a suitable location on their machine and extract the files before they can view the documentation successfully. You can include this information in your email using the following sentence:

"Please save the attached ZIP file to your hard drive and extract the contents of the ZIP file."

4.18.9 Linking to generated HTML documentation

With a little effort you can link existing web pages, from an intranet or web server, to *XSOL Mapping's* generated HTML (HyperText Markup Language) documentation (specifically the "One page website" option). This allows you to quickly link from your existing web pages to relevant parts of your *XSOL Mapping* model documentation. You only need to create these links once, as the link anchors are the same every time the documentation is generated.

You need the following in order to create these links:

1. Some knowledge of HTML. Specifically, you need to know how to create links between one HTML page and another.
2. An HTML editor. If you feel comfortable with editing HTML by hand, this can be as simple as a plain text editor (such as Notepad).

To link to generated HTML documentation:

1. Generate the "One page website" documentation for the required *XSOL Mapping* model. Make sure that the location that you save the documentation to is accessible by the web page which will link to this documentation.
2. Open the web page that you want to link from and insert a link to the generated HTML page. This links you to the top of the generated HTML documentation. If this is all you want to do, then you are finished with this procedure.
3. To link to a specific heading in the HTML, open the generated HTML file in an HTML or text editor.
4. Find the link in the "Table of Contents" to the section that you want to link to.
5. Copy and paste the link to your HTML page. The finished link is in the following form: `link text`.

4.18.10 Using Word 2003 custom sections

If you have Word 2003 installed you can create custom sections in a specific template that you can include when generating the documentation for this layout. These sections (defined as bookmarks in the template) can contain any amount of information and can be inserted one or more times into any place in the document layout.








Custom sections that are inserted into a document layout in some locations will be repeated in the generated documentation. For example, inserting a custom section at the start of the **Processes** section will be repeated for every printed process.



If you encounter problems using this procedure when you are copying and pasting content from web pages or other Word documents, try copying and pasting using the **Edit > Paste Special...** menu option and select the **Unformatted Text** option. This will remove any hidden code that may interfere with the custom section, but you will have to re-apply any formatting that you may have lost.

To insert, create, or remove a Word 2003 custom section:

1. Follow one or more of the following instructions:
 - (a) To **insert** a custom section, drag and drop the section from the **Custom sections** area on the right into the **Sections** area on the left. The list of available sections is dependant on the template selected from the **Word Template** drop-down list.
 - (b) To **create** a custom section, open the template that you want to add a custom section to in Microsoft Word and go to the next step in this procedure.
 - (c) To **remove** a custom section reference from this layout, click on the required custom section in the left area and click on the  button above this area.
To **remove** a custom section reference from the template, open the template in Microsoft Word, click on the **Insert >Bookmark...** menu item, select the required bookmark and click on the **Delete** button.
2. Add the text that you want to use as a custom section after the "ADD CUSTOM SECTIONS AFTER THIS LINE" text in the **Custom sections** section.
3. You need to turn on the **Show All** view to view the paragraph markers in the template. Click on the  button to turn on this view, if it is not on already.
4. Select all the text that you want to include. You **MUST** include the last paragraph marker (the  icon) for the text in order to bookmark it correctly.
 You can not include the last paragraph for the entire document in the bookmark - you always need an empty paragraph at the end of the document that is not part of any bookmark.
5. Click on the **Insert >Bookmark...** menu item.
6. Enter the name of the bookmark in the **Bookmark name** field and click on the **Add** button.
 Bookmark names must begin with a letter and can contain numbers, but no spaces or punctuation characters are allowed. Use the underscore character instead of a space - all underscore characters will be converted to spaces in the generated documentation. Bookmark names starting with "XSOL_" are reserved for XSOL use only and should not be deleted.

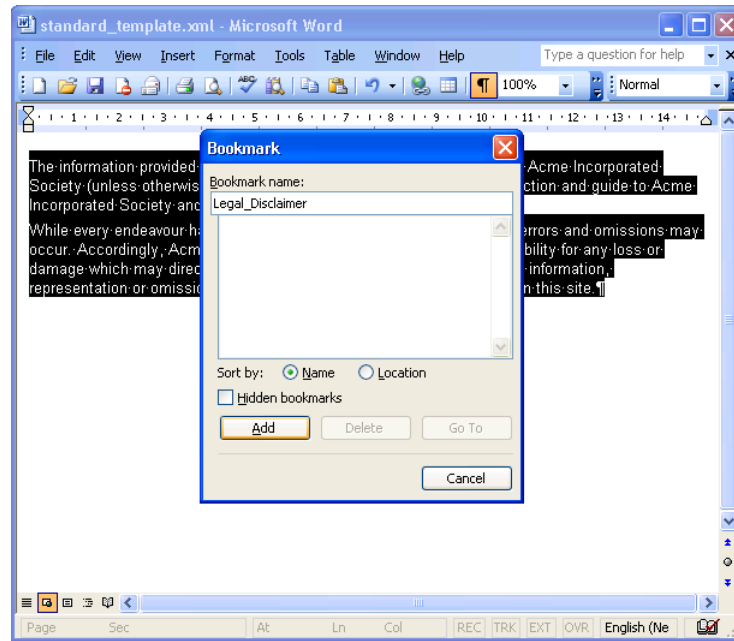


Figure 4.123: Adding a custom section in Word 2003

7. Save and close the Word template. You will need to close the **Layout Editor** window and start generating documentation again in *XSOL Mapping* to make the new custom section available.



If a custom section is invalid for some reason and saved in a documentation layout, it will appear as red in *XSOL Mapping*. To change this you will need to close the **Documentation Generation** window, delete the original bookmark, make any changes to the section that are required, and re-add the bookmark. The custom section will now display as valid in the documentation layout.

4.18.11 Using Word 2003 documentation templates

If you have Word 2003 installed you need to select a template for each document layout that uses Word. The default templates included in the *XSOL Mapping* installation, **standard.template.xml** and **tabular.template.xml**, include a large number of defined styles, the same as the style template included for Word 2000/XP file generation, **xsol.dot**. However, there are several features available in the Word 2003 template that are not available for Word 2000/XP:

- You can change page size and margins.
- You can define headers and footers for each section of the generated documentation, including inserting images.
- You can add custom sections (refer to page 162 for more information).
- You can change individual styles (also available for Word 2000/XP - refer to page 165 for more information).

All changes that you make in the template to the information around the main body of the document (page margins, styles, etc.) will be included in generated Word documentation. Changes made to the main body of the template (all text, images, tables etc. that are not in the header or footer) will be ignored (except for bookmarked custom sections).

To use a Word 2003 documentation template:

1. Copy the default template **standard_template.xml** or **tabular_template.xml** in the **Document template folder** to the same directory (refer to page 175 for more information). Rename the template to a meaningful title.



The **tabular_template.xml** template is used for the **Tabular Word document (processes only)** generated document, and the **standard_template.xml** template is used for all other Word generated documents.

2. Open the new template in Word 2003.
3. Please follow the instructions in the template. Some of the items you can change are as follows:
 - To change **page sizes or margins**, click on the **File >Page Setup...** menu item. Edit any required settings, then click on the **OK** button.

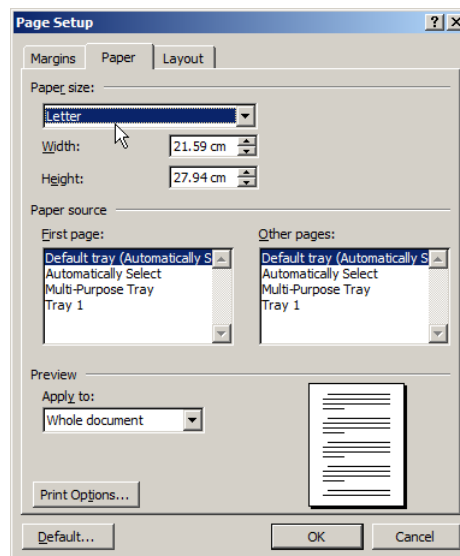


Figure 4.124: Changing a Word 2003 documentation template

- To change **headers or footers**, click on the **View >Header and Footer** menu item. Edit any required settings, then click on the **Close** button.
 - To **add a custom section**, refer to page 162 for more information.
 - To change **one or more styles**, refer to page 165 for more information and start at step 4.
4. Save the document.
 5. Generate a Word document using this template to make sure the changes you have made work correctly.

4.18.12 Using styles with the Microsoft Word output



Refer to page 160 for more information about changing style information in the generated HTML (HyperText Markup Language) output.

Each paragraph in the generated Microsoft Word output has a specific style assigned to it. For example, the main title on the title page is assigned the "TitlePage" style. This allows you to change the formatting of a particular style in the entire document very quickly.

There are two ways you can change styles with Microsoft Word:

- **In the document** - Change the style definition in the generated document. This will change all examples of that style within the document. This is useful for limited, ad-hoc changes.
- **Apply a template (2000/XP)** - Create a template with pre-defined styles of the same name as the generated styles and automatically apply the template to the generated document. The *XSOL Mapping* installation includes a Word template, **xsol.dot**, which is used in the document generation Process and has all the styles named and defined.

xsol.dot can be stored in two locations: the **User templates** location defined in Word, or in the **Document template folder** directory (refer to page 175 for more information).



Refer to page 164 if you need more information about templates for Word 2003.



Changing the style information in generic.css does **not** change User Note styles. All User Note style information is stored either in the model or in **Normal.dot** - refer to page 147 for more information.

To change one style in a generated Word document:

1. Click on the paragraph that you want to change.
2. If you are using Word 2000:
 - (a) Click on the **Format > Styles...** menu option.
 - (b) Alter the style using the available functions. Please refer to your Word online help for more information about changing Word styles.

If you are using Word 2002/2003:

- (a) Click on the **Format > Styles and Formatting...** menu option.
- (b) Click on the down arrow in the **Formatting of selected text** drop-down list and click on the **Modify...** option.
- (c) Alter the style using the available functions. Please refer to your Word online help for more information about changing Word styles.

To change any number of styles for Word 2000/XP using a template:

1. Make a backup of the **C:\Program Files\XSOL\Mapping\Samples\xsol.dot** file to a suitable location.
2. Open *XSOL Mapping* and generate the Word output for a model.
3. Open the generated Word output.
4. If you are using Word 2000:
 - (a) Click on the text that you want to alter. This displays the style name in a drop-down list at the top of the window.
 - (b) Click on the **Format > Styles...** menu option.
 - (c) Alter the style you want to change using the available functions. Please refer to your Word online help for more information about changing Word styles.

If you are using Word 2002:

- (a) Click on the **Format > Styles and Formatting...** menu option.
- (b) Click on the text that you want to alter.

- (c) Alter the style using the available functions. Please refer to your Word online help for more information about changing Word styles.
5. Click on the **Add to template** tick box when you have finished altering the style. This will update **xsol.dot** with the style change when you save the document.
6. Repeat the above steps as many times as required to change various styles. Make sure you always check that the **Add to template** tick box is selected with each change.
7. Save the document to update **xsol.dot**.
8. Generate a Word document again to make sure the changes you have made work correctly.


To change any number of styles for Word 2003 using a template:

1. Make a backup of the template you want to edit to a suitable location (refer to page 164 for more information).
2. Open the Word 2003 template.
3. Click on the **Format > Styles and Formatting...** menu option.
4. Alter the required style using the available functions. Please refer to your Word online help for more information about changing Word styles.
5. Repeat the above steps as many times as required to change various styles.
6. Save the template.
7. Generate a Word document again to make sure the changes you have made work correctly.

4.18.13 Printing tips

This section describes a number of different tips for generating clear and accurate documentation.

Page layout

You can set page layout for Process Flow diagrams by turning on the Page Boundaries option (click on the  button). This allows you to view the diagram in relation to the page.

Diagrams

The HTML (HyperText Markup Language) documentation automatically generates maximum-size diagrams. Word documentation, however, will cut up a large diagram into page-size chunks. There are several Process Flow diagram settings that can help you produce effective Word documents. Please experiment with these settings to see what produces the best results for you.



These options are ignored for HTML and spreadsheet output.

- **Diagram Magnification** - Sets the diagram size as a percentage. For most diagrams 70% - 80% provides excellent readability while displaying the maximum amount of information on the page.
- **Paper Size** - Sets where the diagram will be cut, based on pre-set paper sizes. This should be the same size as your output document for best results. However, you can use the Custom Size option to set exactly the height and width that you required.
- **Rotation for flow diagrams** - Sets the logical "up" position for the diagram. If you are sitting on a chair, the **Portrait** option sets the "up" position towards the ceiling, while **Landscape** sets the "up" position to your left. These settings do not have anything to do with paper orientation - that is controlled by the documentation option chosen, such as **Microsoft Word (portrait)**.

Large diagrams split across multiple pages from left to right and from top to bottom. For example, a large landscape diagram that covers four pages will print in this order:

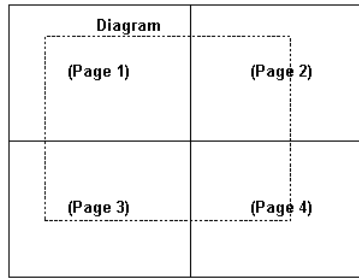


Figure 4.125: Printing a large diagram in landscape mode

The same diagram in portrait mode would print like this:

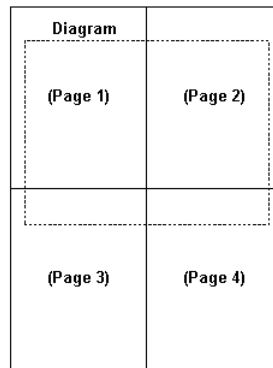



Figure 4.126: Printing a large diagram in portrait mode

Paper size

The paper size for the Microsoft Word output is set at the model properties level.

1. Click on the  icon in the **Design Enterprise** window.
2. Choose your required paper size from the **Paper Size** drop-down list in the **Properties Editor**.

Title page

The following settings influence the appearance of the title page of the generated documentation:

- The company name that appears on the title page varies depending on the following criteria:
 - If there is only one Trading Entity the name of that Trading Entity is used.
 - If there is more than one Trading Entity the name of the Enterprise is used.

4.19 Options

The Options window allows you to set a variety of preferences about how the system runs. Each tab in the Options window is described in the following sections.

4.19.1 General - options

You can change the following preferences in the General tab in the Options window:

- **Automatically open last model** - Opens the most recently saved model when logging on to *XSOL Mapping*. If not ticked, a window opens with a variety of option (refer to page 22 for more information).
- **Recently used model list** - Enter the number of previously used models that you want to see in the **Window** pull-down menu. The models are listed from most-recently used to oldest in the menu.
- **Always backup the model before saving** - This creates an `_Backup\` directory in the same folder as the saved model. Refer to page 37 for more information about backing up and restoring your models.
- **Number of undo operations** - Enter the number of changes that are available using the **Edit > Undo** menu option. You can enter as large a number as you want. However, large numbers of undo items will significantly affect available memory on your machine over time.
- **Default images folder** - This directory stores all images that are used in User Notes to a central location (refer to page 145 for more information.). Change the path in this field or use the **Browse...** button to navigate to the required directory so that all images are available to all users of a model, irrespective of the machine they are using.



The path needs to be a full network path, as drive mappings may not work correctly for all users.



If you change this path to a new directory, all existing images in User Notes are maintained in the same location. This means that, over time, the images in the User Notes in your model(s) will be stored in multiple locations (all previous directories and the current directory).

- **Root document location folder** - Used when certifying Script Tasks that include Document Locations. Sub-directories within the root folder are created automatically using the Document Location's **External Id** property. Refer to page 221 for more information.
- **Annotation background colour** - Click on the required background color for annotation nodes from the drop-down list. This changes the background colour of all annotation nodes.

To set general options:

1. Click on the **Tools > Options...** menu option.
2. Set the options as required.
3. Click on the **OK** button when you have finished setting your options.

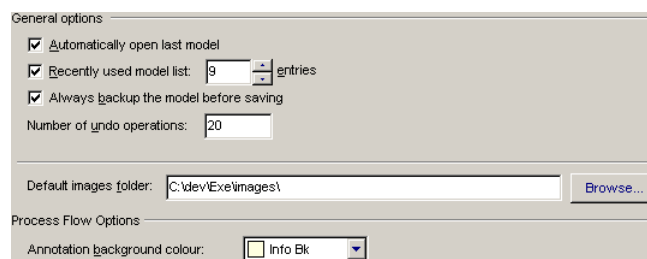


Figure 4.127: Setting General options

4.19.2 New Model Defaults - options

The following options apply to new models when they are created in *XSOL Mapping*:

- **Default caption for new 'True' links** - Enter the default caption for all true links from Condition nodes - refer to page 85 for more information. This only applies to links when they are created - it will not change existing link captions.
- **Default caption for new 'False' links** - Enter the default caption for all false links from Condition nodes - refer to page 85 for more information. This only applies to links when they are created - it will not change existing link captions.
- **Draw 'True' links first?** - If ticked, true links are created first from Condition nodes - refer to page 85 for more information.
- **Default stage flow 'Print in documentation'** - If ticked, Task Process Flow is automatically included in generated documentation. This only applies to Stages when they are created - it will not change existing Stages.
- **Default process flow 'Show role assignments'** - If ticked, Role information is displayed on Process Flow diagrams - refer to page 96 for more information.
- **Default 'True' value** - Optional. Enter the default true value for all Binary data types (Variables and User Notes - refer to page 128 and page 145 for more information).
- **Default 'False' value** - Optional. Enter the default false value for all Binary data types (Variables and User Notes - refer to page 128 and page 145 for more information).
- **Default 'Null' value** - Optional. Enter the default unknown value for all Binary data types (Variables and User Notes - refer to page 128 and page 145 for more information).

To set model options:

1. Click on the **Tools >Options...** menu option.
2. Click on the **New Model Defaults** tab.
3. Set the options as required.
4. Click on the **OK** button when you have finished setting your options.

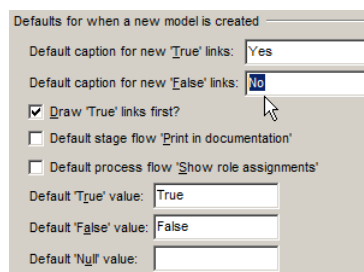


Figure 4.128: Setting New Model Defaults options

4.19.3 Highlighting - options


You can set a number of highlighting preferences, including whether or not invalidate items are highlighted and the highlighting appearance of regular items. You can also choose whether invalidate items includes or excludes warnings.

There are three types of highlighting items:

- **Normal colour** - Normal appearance of the item.
- **Selected colour** - Focus has been set on the item (the item has been clicked on).
- **Unfocused colour** - A sub-entry of the focused item has been clicked on.

You can set the colour and style of each of these items.

To set highlighting options:

1. Click on the **Tools >Options...** menu option.
2. Click on the **Highlighting** tab.
3. Tick the **Display incomplete entities using the following style** tick box if you want to turn on the highlighting for incomplete entities in the model.
-  Tick the **Include warnings** tick box if you want to include warnings as highlighted items.
4. Tick the **Display cut/copied entities using the following style** tick box if you want to turn on the highlighting for cut or copied entities in the model.
5. For incomplete items or regular highlighting you can do the following:
 - Set the normal, focused, and unfocussed colour for each item using the available drop-down lists.
 - Set the style of each item using the **Style** drop-down list.
 - Set the item to appear as underlined or struck out using the **Underline** and **Strikeout** tick boxes.
6. Tick the **Display highlighted entities in bold font** tick box if you want *XSOL Mapping* to display highlighted entities in a bold font.
7. Tick the **Display deleted entities in strikeout font** tick box if you want *XSOL Mapping* to display deleted entities in a strikeout font.
8. Click on the **OK** button when you have finished setting your options.

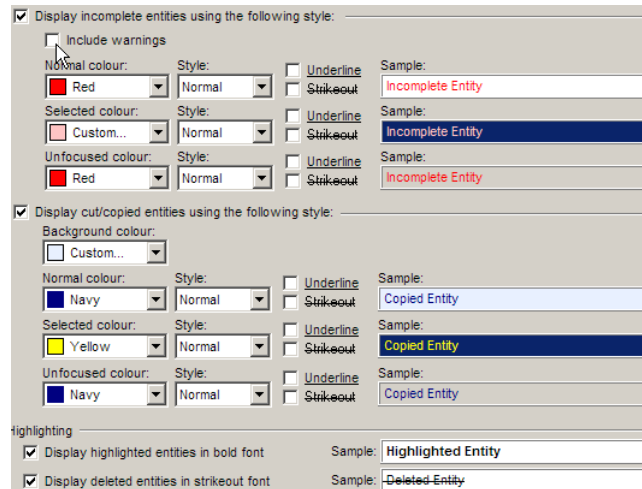


Figure 4.129: Setting highlighting options

4.19.4 Properties Editor - options

You can set the visual appearance used for the text in the **Properties Editor**. The following areas can be configured:

- **Normal properties** - Applies to default properties in the **Properties Editor**. Default properties are all properties that do not match any other property definition (such as error or read-only properties).
- **Grouped - different** - Applies to default properties in the **Properties Editor** when more than one item has been selected.
- **Sub-property** - Applies to indented text, properties that are contained within other properties.
- **Sub-prop - different** - Applies to indented text when more than one item has been selected.
- **Error property** - Applies to invalid properties.
- **Error - different** - Applies to invalid properties when more than one item has been selected.
- **Read-only property** - Applies to read-only properties.
- **Read-only - different** - Applies to read-only properties when more than one item has been selected.

Refer to page 27 for more information about the **Properties Editor**.

To set options for the Properties Editor:

1. Click on the **Tools >Options...** menu option.
2. Click on the **Properties Editor** tab.
3. Select the colour that you want to use for a particular style from the first drop-down list.
4. Select the style type (such as **Bold** or **Italics**) from the **Style** drop-down list.
5. Tick the **Underline** or **Strikeout** tick boxes, if required. You can view a sample of the altered text in the **Sample** area.

- Repeat the above steps for as many property types as you want to change.

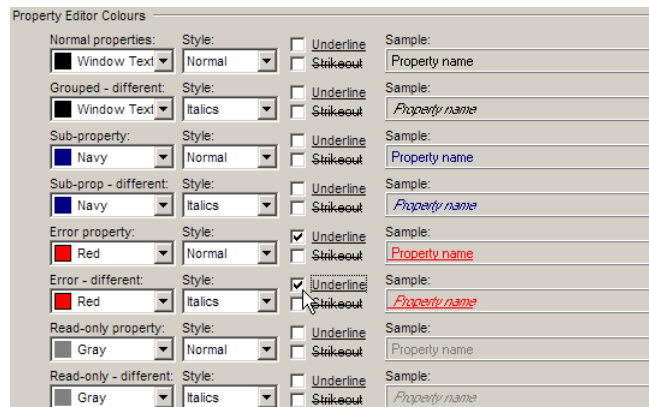


Figure 4.130: Setting options for the Properties Editor

- Click on the **OK** button when you have finished setting your options.

4.19.5 Icons - options

Each function in the **Icons** tab is described in the following sections.

Displaying Task icons

XSOL Mapping enables you to show or hide the default icons shipped with *XSOL Mapping*.

To display or hide Task icons

- Click on the **Tools > Options...** menu option.
- Click on the **Icons** tab.
- If you need to hide any of the Task icons, select the **Hide these icons** tick box, and tick the icons that you do not want displayed.

Remove the tick from the **Hide these icons** tick box if you do not want to hide any Task icons.

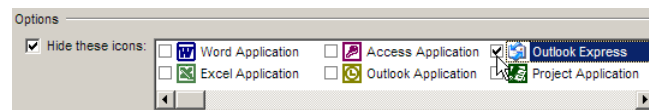


Figure 4.131: Hiding a specific Task icon

Adding additional icons

You can add new icons that can be chosen by **Non-XSOL** Task types. *XSOL Mapping* supports 24-bit bitmap icons - you will need some sort of screen capture and/or image editing software to produce these graphic files.



Custom Task icon images are stored in the **Default images folder** - refer to page 169 for more information.

To add new Task icons:

1. Locate or create the icons that you want to use. You need to create two icons for each new icon that you want to use, one 32x32 pixels and one 16x16 pixels. All icons must be 24-bit bitmaps.



The top-left pixel in the icon is the default colour.

2. Click on the **Tools >Options...** menu option.
 3. Click on the **Icons** tab.
 4. Click on the **Add** button.
 5. Enter the name of the Task icon in the **Description** field.
 6. Select the 16x16 pixel bitmap in the **Small icon** field.
 7. Select the 32x32 pixel bitmap in the **Large icon** field.
- Both icons should be displayed in the preview area.
8. Click on the **OK** button. The icon is now available for use.

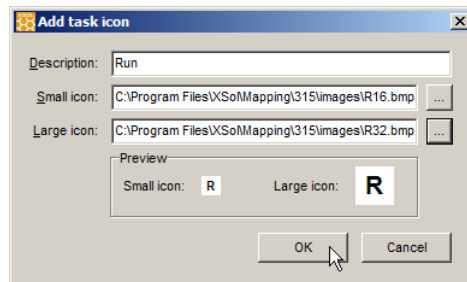


Figure 4.132: Adding a new Task icon


Using Task icon associations

You can set *XSOL Mapping* to display certain types of icons automatically based on a variety of criteria. The following association options are available:

- **Document Type** - Creates an icon association based on the target file's file type. For example, an Excel document links the chosen icon to the **.xls** file type.
- **Internet Protocol** - Creates an icon association based on the start of the target file's path. For example, an Internet URL links the chosen document to the **http** portion of the URL.
- **Application** - Creates an icon association based on the target file, if it is an application (.exe). For example, the Notepad application links the chosen icon to the target file **notepad.exe**.

To add or remove a Task icon association:

1. Click on the **Tools >Options...** menu option.
2. Click on the **Icons** tab.
3. If you need to turn the icon association functionality on or off, click on the following tick boxes:

- **Enable Associations** - Turns the Task icon association functionality on and off. The default setting is on.
 - **Automatically add to associations** - Automatically adds new Task icon associations to the associations list based on user actions (such as choosing a new Task icon for a particular Non-XSOL Task). The default setting is on.
4. To remove an association, click on the required association in the list and click on the **Remove** button.
 5. To add an association, click on the **Add** button.
-  It is usually easier to enter an association through assigning a particular icon to a task with a specific target. Continue with the procedure only if you want to enter an association manually.
6. Click on the type of association from the **Type** drop-down list. Refer to the description of these types at the beginning of this topic.
 7. Enter the value for the association in the **Protocol** field. For example, for an Internet URL enter **HTTP**.
 8. Click on the required icon from the **Icon** drop-down list.
 9. Click on the **OK** button when you have finished setting your associations.

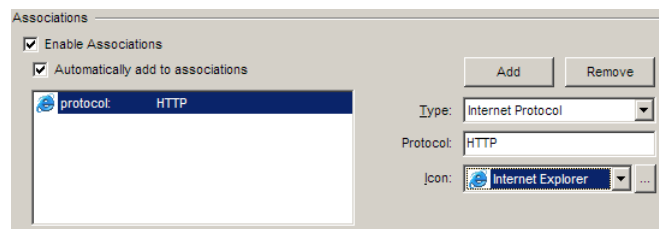


Figure 4.133: Adding a Task icon association

4.19.6 Documentation - options

You can set the location where the documentation layouts and documentation templates are stored. (By default these are in sub-directories of the **C:\Documents and Settings\All Users\Application Data\XSOL** directory.) Both settings are mandatory - you can not generate any documentation without them.



The other parts of this tab allow you to add or edit the documentation output formats that are included in *XSOL Mapping*. Do not alter the properties for this module unless you know what you are doing.

To set the documentation layout and template directories:

1. Click on the **Tools > Options...** menu option.
2. Click on the **Documentation** tab.
3. Use the browse button to select the required directory for the **Documentation layout folder** and/or **Documentation template folder** fields.

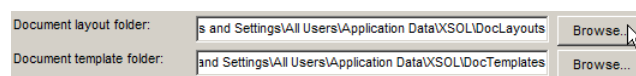


Figure 4.134: Changing the documentation layout directory



If you change the documentation template directory, you need to copy the contents of the old directory to the new one.



If you change the documentation layout directory, you only need to copy the contents of the old directory to the new one if you need to keep your defined layouts. If you do not need to keep them, *XSOL Mapping* will re-create the default layouts the next time you generate documentation.

4.19.7 Predefined Filters - options

You can set up one or more predefined filters in your model. These filters are used in a running *XSOL Automation* system and allow you to limit what sort of files a user can select. For example, you can set up a filter that allows Users to only select Microsoft Word documents (refer to page 67 for more information about how predefined filters are used in *XSOL Automation*).

To add predefined filters:

1. Click on the **Tools >Options...** menu option.
2. Click on the **Predefined Filters** tab.
3. Enter the description of the required filter in the **Name** column and the filter itself in the **Filter** column. You can use the asterisk character (*) as a wildcard - for example, a filter to only choose Microsoft Word documents would be *.doc.
4. Repeat the above step for as many filters as you require. You can create a new filter line by pressing the **down arrow** key, or remove an empty line by pressing the **up arrow** key.

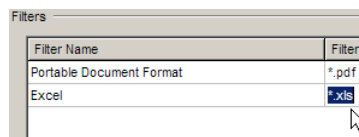


Figure 4.135: Adding a predefined filter

5. Click on the **OK** button when you have finished setting your options.

4.19.8 Editor Options - options

You can set various options on *XSOL Mapping's Expression Editor*, including font type and size, tab indentation size, and line numbering. The following options are available:

- **Font name** - Choose the font used to display Expressions.
- **Size** - Choose the font size used to display Expressions.
- **Auto indent** - Automatically indents a new line to the same indentation level as the previous line.
- **Python indent** - Automatically indents Expressions according to Python indentation rules. For example, after a colon (:) the next line will be indented by one tab more than the original line.
- **Smart tab** - Automatically indents a line to the same indentation level as the previous line when you press the Tab key.
- **Tab to spaces** - Converts all tabs to spaces.

- **Tab stops** - Enter the number of spaces which are inserted every time the Tab key is pressed.
- **Drag-N-drop editing** - Allows you to drag operators and parameters onto a script.
- **Insert mode** - Allows you to insert text where the cursor is placed without overwriting any already existing text.
- **Group undo** - Sets each undo action (Ctrl+Z) to undo an entire word, not just a keystroke.
- **Undo limit** - Sets the number of undo actions that are available from the **Expression Editor**.
- **Show gutter** - Displays a gray strip of space on the left side of the **Expression Editor**.
- **Gutter width** - Sets the width of the gutter in pixels.
- **Show line numbers** - Displays line numbers in the gutter.
- **Show right margin** - Displays a right-hand margin in the **Expression Editor**.

To set general editor options:

1. Click on the **Tools >Options...** menu option.
2. Click on the **Editor Options** tab.
3. Set the options as described at the beginning of this topic. Many options will update the display area automatically so that you can see if they are acceptable.
4. Click on the **OK** button when you have finished setting your options.

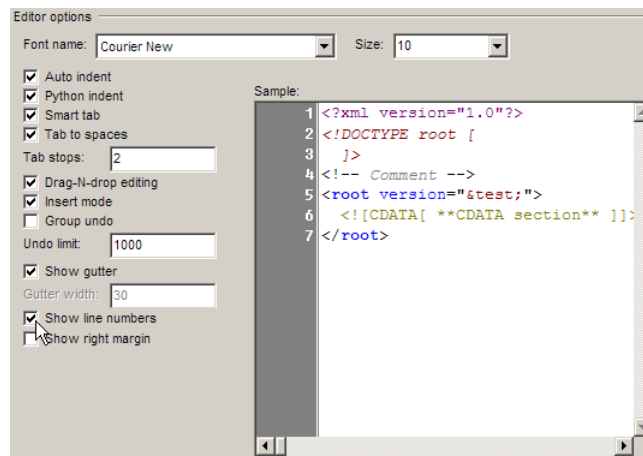


Figure 4.136: Setting Editor options

4.19.9 Editor Colours - options

You can set various colour options for *XSOL Mapping's* **Expression Editor** and save them as a colour scheme, if required. The following options are available:

- **Colour scheme** - Displays a list of colour schemes. Each element in the scheme has an associated colour.
- **Language** - Choose the XSOL Python Script option for changing the **Expression Editor**.

- **Element** - Click on the required item. All following items options apply to the chosen element.
- **Foreground** - Click on the foreground colour from the drop-down list.
- **Background** - Click on the background colour from the drop-down list.
- **Text style** - Click on the listed tick boxes to change the appearance of the chosen element.

To set editor colour options:

1. Click on the **Tools >Options...** menu option.
2. Click on the **Editor Colours** tab.
3. Choose a colour scheme, if required. If this is all you need to do in this window, click on the **OK** button, and you are now finished. If you need to set particular colour options, continue with this procedure.
4. Choose the XSOL Python Script option from the **Language** drop-down list.
5. Click on the required element from the **Element** list.
6. Set the various colour options, as described at the beginning of this topic.
7. Repeat the last two steps for as many elements as required.
8. If you want to save these settings as a particular colour scheme, click on the **Save As** button, enter the name of the colour scheme, and click on the **OK** button. You can now choose this colour scheme from the **Colour scheme** drop-down list whenever required.
9. Click on the **OK** button when you have finished setting your options.

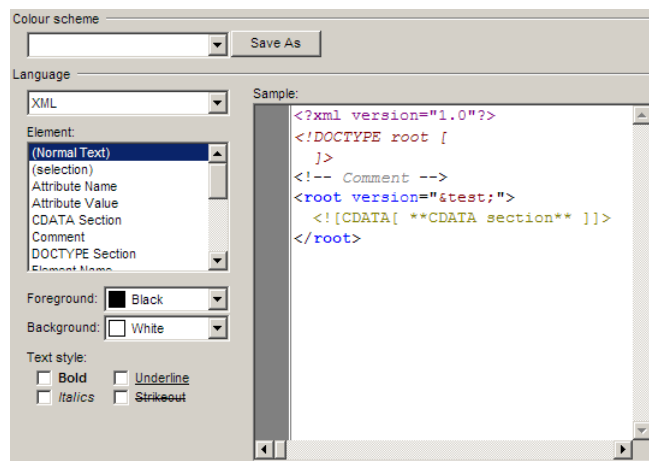


Figure 4.137: Setting Editor Colour options

4.19.10 Default Icons - options

You can set the default icons for any Process Flow icon. You can change the default icon used by any Process Flow node and you can save these changes as the new default icons for the application. This allows you to visually standardize your Process Flow diagrams. Refer to page 92 to change all the icons for a specific model or page 91 if you only want to change the appearance of one individual node.

To set default icons:

1. Click on the **Tools >Options...** menu option.
2. Click on the **Default Icons** tab.
3. Click on the node that you want to change and click on the **Change** button.

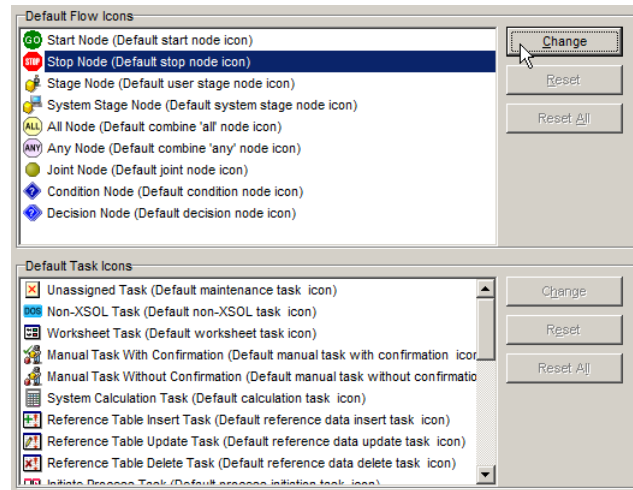


Figure 4.138: Setting Default Icons options

4. Click on the icon that you want to use for this node and click on the **OK** button.
5. Click on the **OK** button when you have finished setting your options.

4.20 Keyboard shortcuts

GENERAL:

- **Ctrl+N** - Create a new model
- **Ctrl+O** - Load a model
- **Ctrl+S** - Save the current model
- **Shift+Ctrl+S** - Save the current model under another name
- **Alt+F4, Alt+X** - Close *XSOL Mapping*
- **Ctrl+X, Shift+Del** - Cut the selected entity
- **Ctrl+C, Ctrl+Ins** - Copy the selected entity
- **Ctrl+V, Shift+Ins** - Paste the entity in the clipboard
- **Shift+Ctrl+V** - Paste the linked entity in the clipboard
- **Ctrl+A** - Select all the text in text field
- **Ctrl+Z** - Undo the last action
- **Ctrl+Shift+Z** - Redo the last action
- **Ctrl+E** - Display the **Design Enterprise** window
- **F11** - Open or close the **Properties Editor**
- **Shift+F11** - Roll up or down the **Properties Editor**
- **Ctrl+R** - Generate documentation
- **Ctrl+M** - Validate the current model
- **Ctrl+K** - Validate the selected entity
- **Ctrl+W** - Check where the selected entity is used
- **Ctrl+T** - Open the Options window
- **F8** - Open or close the **Customise** window
- **Shift+F8** - Roll up or down the **Customise** window
- **Alt+left arrow** - View the previous tab
- **Alt+right arrow** - View the next tab
- **Del** - Delete the selected item

HELP:

- **F1** - Open the online help (front page)
- **Shift+F1** - Open the online help (index tab)
- **Ctrl+Shift+F1** - Open the online help (search tab)

DESIGN ENTERPRISE WINDOW:

- **Alt+N** - Add a new Trading Entity
- **Shift+Alt+P** - Open the Design for Mapping window
- **Shift+Alt+D** - Open the Design For Automation window

Roles

- **Alt+L** - Add a Role

Macro Expressions

- **Alt+M** - Add a Macro Expression

Logical Printers

- **Alt+P** - Add a Logical Printer

Document Locations

- **Alt+D** - Add a Document Location

DESIGN BUSINESS PROCESSES WINDOW:

Processes

- **Alt+G** - Add a Process Group
- **Alt+P** - Add a Process
- **Alt+C** - Add a Companion Process
- **Alt+S** - Add a Stage
- **Alt+K** - Add a Task
- **Alt+A** - Add an Escalation Level
- **Alt+L** - Add a Role
- **Shift+Alt+P** - Open the **Design Process Flow** window for a Process
- **Shift+Alt+S** - Open the **Design Process Flow** window for a Stage

Tasks

- **Alt+K** - Add a Task
- **Alt+O** - Add a Worksheet
- **Alt+M** - Add an attachment (Send Email Tasks only)
- **Alt+R** - Add a Stored Variable
- **Alt+V** - Add a Work Variable
- **Alt+N** - Add an Expression
- **Alt+B** - Add a Business Rule

- **Alt+L** - Add a Role
- **Alt+L** - Add a limit to location
- **Shift+Alt+O** - Open the Design Worksheet Layout window
- **Ctrl+Alt+W** - Make a Worksheet common or not
- **Ctrl+Alt+N** - Assign an Expression
- **Ctrl+Alt+B** - Assign a Business Rule

Variables Used

- **Alt+G** - Add a Process Group
- **Alt+P** - Add a Process
- **Alt+C** - Add a Companion Process
- **Alt+S** - Add a Stage
- **Alt+K** - Add a Task
- **Alt+O** - Add a Worksheet
- **Alt+R** - Add a Stored Variable
- **Alt+V** - Add a Work Variable
- **Alt+N** - Add an Expression
- **Alt+U** - Assign a Business Rule
- **Alt+D** - Add a Document Location
- **Alt+L** - Add a Role
- **Alt+Y** - Assign an Expression
- **Shift+Alt+P** - Open the **Design Process Flow** window for a Process
- **Shift+Alt+S** - Open the **Design Process Flow** window for a Stage
- **Shift+Alt+O** - Open the Design Worksheet Layout window
- **Ctrl+Alt+W** - Make a Worksheet common or not
- **Ctrl+Alt+B** - Assign a Business Rule
- **Ctrl+Alt+R** - Open the **Process Variable Map Editor** window

To Do Lists Button

- **Alt+G** - Add a Process Group
- **Alt+P** - Add a Process
- **Alt+C** - Add a Companion Process
- **Alt+S** - Add a Stage
- **Shift+Alt+P** - Open the **Design Process Flow** window for a Process

- **Shift+Alt+S** - Open the **Design Process Flow** window for a Stage
- **Shift+Alt+R** - Open the **Sort Order Editor** window

Process Summaries

- **Alt+G** - Add a Process Group
- **Alt+P** - Add a Process
- **Alt+C** - Add a Companion Process
- **Alt+S** - Add a Process Summary
- **Alt+O** - Add a Worksheet
- **Alt+V** - Add a Work Variable
- **Alt+N** - Add an Expression
- **Alt+R** - Add a Filter
- **Alt+L** - Add a Role
- **Shift+Alt+P** - Open the **Design Process Flow** window for a Process
- **Shift+Alt+S** - Open the **Design Process Flow** window for a Stage
- **Shift+Alt+R** - Open the **Sort Order Editor** window
- **Ctrl+Alt+W** - Make a Worksheet common or not
- **Ctrl+Alt+N** - Assign an Expression

Variables

- **Alt+G** - Add a Variable Group
- **Alt+R** - Add a Stored Variable
- **Alt+V** - Add a Work Variable
- **Alt+N** - Add an Expression
- **Alt+B** - Add a Business Rule
- **Alt+L** - Add a Role

Ref Variables

- **Alt+N** - Add an Expression
- **Alt+B** - Add a Business Rule

DESIGN PROCESS FLOW WINDOW:

- **Esc** - Chooses the Select tool
- **Alt+R** - Select the Start tool
- **Alt+P** - Select the Stop tool

- **Alt+S** - Select the Stage tool
- **Alt+K** - Select the Task tool
- **Alt+C** - Select the Condition tool
- **Alt+D** - Select the Decision tool
- **Alt+O** - Select the Fork tool
- **Alt+L** - Select the All tool
- **Alt+Y** - Select the Any tool
- **Alt+J** - Select the Joint tool
- **Alt+N** - Select the Annotation tool
- **Alt+M** - Select the Swim Lane tool
- **Ctrl+Alt+R** - Adds Go and Stop nodes
- **Alt+N** - Adds a Condition (in the **Decision Table** panel)
- **Alt+D** - Adds a Decision (in the **Decision Table** panel)
- **Shift+Alt+N** - Display the Expression (in the **Decision Table** panel)
- **Ctrl+Alt+O** - Rotate a Fork
- **Ctrl+Alt+C** - Swap the Yes/No links on a Condition node
- **Ctrl+Alt+U** - Automatically link all nodes
- **Ctrl+Alt+N** - Validate the Process Flow links
- **Shift+Alt+X** - Display or hides the Expansion Panel.
- **Shift+Alt+B** - Display page boundaries for the Process Flow diagram
- **Shift+Alt+G** - Display grid lines
- **Alt+NumPlus/Shift+Alt+Equal** - Zoom in
- **Alt+NumMinus/Shift+Alt+Underscore** - Zoom out
- **Alt+NumAsterisk/Shift+Alt+8** - Return to 100% view
- **Shift+Alt+P** - Toggle snap to grid
- **Alt+Up arrow** - Move panel up
- **Alt+Down arrow** - Move panel down
- **F4** - Review Stage Process Flow to print in documentation
- **F5** - Show Companion Processes available for Stages
- **F6** - Show available Processes
- **F7** - Show available Reference Variables
- **F9** - Filter the displayed Stages via Roles
- **F10** - Display Tasks available for Stages

DESIGN WORKSHEET LAYOUT WINDOW

- **Alt+L** - Select the Insert Label tool
- **Alt+M** - Select the Insert Image tool
- **Alt+B** - Select the Insert Bevel tool
- **Alt+G** - Select the Insert Group tool
- **Alt+H** - Select the Insert Horizontal Rule tool
- **Shift+Alt+G** - Display grid dots
- **Shift+Alt+P** - Toggle snap to grid
- **Shift+Alt+1** - Set the grid size as 5x5 pixels
- **Shift+Alt+2** - Set the grid size as 10x10 pixels
- **Shift+Alt+3** - Set the grid size as 15x15 pixels
- **Shift+Alt+4** - Set the grid size as 20x20 pixels
- **Shift+Alt+0** - Display the custom **Grid Size** dialog
- **F10** - Display the **Tab Order** dialog
- **Shift+Alt+Left** - Move all selected widgets in line with the far left widget
- **Shift+Alt+Right** - Move all selected widgets in line with the far right widget
- **Alt+C** - Move all selected widgets in line with the horizontal central point between all widgets
- **Shift+Alt+C** - Move all selected widgets so that they are equally spaced horizontally
- **Shift+Alt+Up** - Move all selected widgets in line with the top widget
- **Shift+Alt+Down** - Move all selected widgets in line with the bottom widget
- **Alt+V** - Move all selected widgets in line with the vertical central point between all widgets
- **Shift+Alt+V** - Move all selected widgets so that they are equally spaced vertically
- **Alt+K** - Send the selected widget to the back
- **Alt+R** - Bring the selected widget to the front
- **Alt+S** - Set the default widget size
- **Alt+D** - Set the Reference Variable for the datafield

DESIGN TABLES WINDOW:**Tables**

- **Alt+G** - Add a Table Group
- **Alt+R** - Add a Resource
- **Alt+N** - Add a Resource Extension
- **Alt+P** - Add a Relationship

- **Alt+C** - Add a Control Group
- **Alt+A** - Add a datafield
- **Alt+S** - Add a Secondary Index
- **Alt+U** - Add a calculated datafield
- **Alt+O** - Add a parent Control Group field
- **Alt+U** - Add a UId field
- **Alt+B** - Add a datafield constraint
- **Alt+D** - Add a Document Location
- **Alt+L** - Add a Role

Maintenance Tasks

- **Alt+G** - Add a Table Group
- **Alt+R** - Add a Resource
- **Alt+N** - Add a Resource Extension
- **Alt+P** - Add a Relationship
- **Alt+C** - Add a Control Group
- **Alt+K** - Add a Maintenance Task
- **Alt+O** - Add a Worksheet
- **Alt+L** - Add a Role
- **Shift+Alt+O** - Open the Design Worksheet Layout window
- **Ctrl+Alt+W** - Make a Worksheet common or not

Control Group entries

- **Alt+Y** - Add a Control Group entry

Lookup Lists

- **Alt+G** - Add a Lookup List Group
- **Alt+O** - Add a Lookup List
- **Alt+D** - Add a Display Field
- **Alt+L** - Add a Role

Expression Editor

- **Enter** - Line break
- **Shift+Enter** - Line break
- **Ctrl+C** - Copy

- **Ctrl+Ins** - Copy
- **Ctrl+X** - Cut
- **Shift+Del** - Cut
- **Ctrl+V** - Paste
- **Shift+Ins** - Paste
- **Tab** - Indents the current line or selected lines
- **Shift+Tab** - Un-indents the current line or selected lines
- **Ctrl+A** - Select all
- **Up** - Move the cursor up one line
- **Shift+Up** - Extend the selection up one line
- **Ctrl+Up** - Scroll the window up one line
- **Down** - Move the cursor down one line
- **Shift+Down** - Extend the selection down one line
- **Ctrl+Down** - Scroll the window down one line
- **Left** - Move the cursor left by one character
- **Shift+Left** - Extend the selection to the left
- **Ctrl+Left** - Move the cursor left by one word
- **Shift+Ctrl+Left** - Select the word to the left
- **Right** - Move the cursor right by one character.
- **Shift+Right** - Extend the selection to the right
- **Ctrl+Right** - Move the cursor right by one word.
- **Shift+Ctrl+Right** - Select the word to the right
- **Del** - Delete the last character to the right of the cursor
- **BkSp**, **Shift+BkSp** - Delete the last character to the left of the cursor
- **Ctrl+BkSp** - Delete the last word to the left of the cursor
- **Ctrl+Shift+0 .. 9** - Toggle a bookmark at the cursor position
- **Ctrl+0 .. 9** - Jump to the bookmark
- **PgUp** - Move the cursor to the next page up
- **Shift+PgUp** - Extend the selection up one section
- **Ctrl+PgUp** - Go to the top of the page
- **PgDn** - Move the cursor to the next page down
- **Shift+PgDn** - Extend the selection down one section
- **Ctrl+PgDn** - Go to the bottom of the page

- **Home** - Go to the start of the line
- **Shift+Home** - Extend the selection to the start of the line
- **Ctrl+Home** - Go to the top of the expression
- **Shift+Ctrl+Home** - Extend the selection to the top of the expression
- **End** - Go to the end of the line
- **Shift+End** - Extend the selection to the end of the line
- **Ctrl+End** - Go to the bottom of the expression
- **Shift+Ctrl++End** - Extend the selection to the bottom of the expression
- **Ctrl+Z** - Undo (unlimited levels)
- **Alt+BkSp** - Undo (unlimited levels)
- **Ctrl+Shift+Z** - Redo (unlimited levels)
- **Shift+Alt+BkSp** - Redo (unlimited levels)
- **Ctrl+F** - Find text
- **Ctrl+R** - Replace text
- **Ctrl+M** - Insert a newline to the left of the cursor
- **Ctrl+N** - Insert a newline to the right of the cursor
- **Shift+Ctrl+I** - Indent a block of text
- **Shift+Ctrl+U** - Un-indent a block of text
- **Ctrl+T** - Delete nearest word
- **Ctrl+Y** - Delete line
- **Shift+Ctrl+Y** - Delete to end of line

TREE VIEW:

- **Ctrl+Home** - Scroll the tree to the top
- **Ctrl+End** - Scroll the tree to the bottom
- **Page Up** - Scroll the tree and selects the node one page up
- **Page Down** - Scroll the tree and selects the node one page down
- **Up arrow** - Select the next node up the tree
- **Down arrow** - Select the next node down the tree
- **Left arrow** - Select the parent node or collapses the current node
- **Right arrow** - Select the first child node or expands the current node
- **Backspace** - Select the parent node
- **+** (**plus**) - Open the closed item
- **-** (**minus**) - Close the opened item

- ***** (**asterisk**) - Open the selected node and all children
- **/** (**forwardslash**) - Close the selected node and all children
- **Shift+Ctrl + (numeric keypad plus)** - Open all items. If you are using a laptop, you will need to turn on your numeric keypad to use this shortcut.
- **Shift+Ctrl - (numeric keypad minus)** - Close all items
- **Ctrl + (plus)** - Resize all columns to fit text
- **Ctrl - _** - Resize all columns back to what they were before **Ctrl + (plus)**
- **Shift+Left arrow** - Select the parent node
- **Shift+Right arrow** - Select the first child node
- **Ctrl+PageUp** - Scroll the tree one page up
- **Ctrl+PageDown** - Scroll the tree one page down
- **Ctrl+Up arrow** - Scroll the tree one line up
- **Ctrl+Down arrow** - Scroll the tree one line down
- **Ctrl+Left arrow** - Scroll the tree to the left
- **Ctrl+Right arrow** - Scroll the tree to the right
- **Shift+Page Up** - Extend the selection to the node one page up
- **Shift+Page Down** - Extend the selection to the node one page down
- **Shift+Up arrow** - Extend the selection one line up
- **Shift+Down arrow** - Extend the selection one line down
- **Ctrl+A** - Select all nodes (except that this currently clashes with another system function)
- **F2** - Put the selected node in the tree view into edit mode
- **Esc** - Stop edit mode

5 Registration details

You can check your *XSOL Mapping* registration details and software expiration date whenever required, often necessary when lodging support calls.

To check your registration details:

1. Click on the **Help > About** menu option.
2. The registration name and expiration date are listed near the top of the window. All *XSOL Mapping* modules are also listed - you may be required to state a module's version number for a particular technical issue.

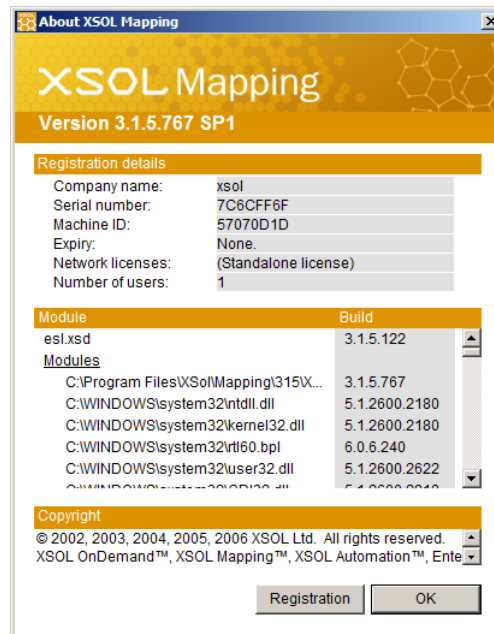


Figure 5.1: Checking registration details

3. Click on the **OK** button when you have finished checking your registration details.

6 End-User License Agreement

IMPORTANT - READ CAREFULLY: This End-User License Agreement ("EULA") is a legal agreement between you (either an individual or a single entity) and **XSOL** Limited, the owner of *XSOL Mapping*. This EULA covers the *XSOL Mapping* software only and includes computer software, the associated media, any printed materials, and any "on-line" or electronic documentation. By installing, copying or otherwise using the *XSOL Mapping* software, you agree to be bound by the terms of this EULA. If you do not agree to the terms of this EULA, **XSOL** is unwilling to license the *XSOL Mapping* software to you. In such event, you may not use or copy the *XSOL Mapping* software, and you should promptly contact **XSOL** for instructions for returning the unused product.

SOFTWARE LICENSE

The *XSOL Mapping* software is protected by copyright laws and international copyright treaties, as well as other intellectual property laws and treaties. **XSOL** owns the title, copyright, and other intellectual property rights in the *XSOL Mapping* software. The *XSOL Mapping* software is licensed, not sold.

GRANT OF LICENSE

This EULA grants you the following rights:

- **Software:** You may install and use one copy of the *XSOL Mapping* on a single computer.
- **Storage/Network Use:** You may not use *XSOL Mapping* over an internal network or distribute the *XSOL Mapping* software to your other computers over an internal network.
- **Back-up Copy:** You may create one (1) back-up copy of the *XSOL Mapping*. You may use the back-up copy solely for archival purposes.
- **Software Transfer:** You may permanently transfer all of your rights under this EULA, provided you retain no copies, you transfer all of the *XSOL Mapping* (including all copies, component parts, the media and printed materials, any upgrades, and this EULA), you provide **XSOL** notice of your name, company, and address and the name, company, and address of the person to whom you are transferring the rights granted herein, and the recipient agrees to the terms of this EULA.
- **Separation of Components:** The *XSOL Mapping* software is licensed as a single product. Its component parts may not be separated for use on more than one computer.

DESCRIPTION OF OTHER RIGHTS AND LIMITATIONS

- **Termination:** Without prejudice to any other rights, **XSOL** may cancel this EULA if you do not abide by the terms and conditions of this EULA, in which case you must destroy all copies of the *XSOL Mapping* software and all of its components parts.
- **Limitation on Reverse Engineering, Decompilation and Disassembly:** You may not reverse engineer, decompile, or disassemble the *XSOL Mapping* software in whole or in part.
- **Rental:** You may not rent or lease the *XSOL Mapping* software.
- **Single Computer:** The *XSOL Mapping* software is licensed for use on a single computer. The *XSOL Mapping* software may only be used on a single computer at a time.

LIMITED WARRANTY

XSOL warrants that the *XSOL Mapping* software will perform substantially in accordance with the accompanying materials for a period of 90 days from the date of receipt. If an implied warranty or condition is created by your state/jurisdiction and federal or state/provincial law prohibits disclaimer of it, you also have an implied warranty or condition, BUT ONLY AS TO DEFECTS DISCOVERED DURING THE PERIOD OF THIS LIMITED WARRANTY (NINETY DAYS). AS TO ANY DEFECTS DISCOVERED AFTER THE NINETY (90) DAY PERIOD, THERE IS NO WARRANTY OR CONDITION OF ANY KIND.

Any supplements or updates to the *XSOL Mapping* software, including without limitation, any (if any) service packs or hot fixes provided to you after the expiration of the ninety-day Limited Warranty period are not covered by any warranty or condition, express, implied, or statutory.

EXCLUSION OF INCIDENTAL, CONSEQUENTIAL, AND CERTAIN OTHER DAMAGES

To the maximum extent permitted by applicable law, in no event shall **XSOL** be liable for any special, incidental, indirect, or consequential damages whatsoever (including, but not limited to, damages for loss of profits or confidential or other information, for business interruption, for personal injury, for loss of privacy, for failure to meet any duty including of good faith or of reasonable care, for negligence, and for any other pecuniary or other loss whatsoever) arising out of or in any way related to the use of or inability to use the *XSOL Mapping* software, the provision of or failure to provide Support Services, or otherwise under or in connection with any provision of this EULA, even in the event of the fault, tort (including negligence), strict liability, breach of contract, or breach of warranty of **XSOL**, and even if **XSOL** or any supplier has been advised of the possibility of such damages.

DISCLAIMER OF WARRANTIES

The Limited Warranty that appears above is the only express warranty made to you and is provided in lieu of any other express warranties (if any) created by any documentation or packaging. Except for the Limited Warranty and to the maximum extent permitted by applicable law, **XSOL** provides the *XSOL Mapping* software and Support Services (if any) AS IS AND WITH ALL FAULTS, and hereby disclaim all other warranties and conditions, either express, implied, or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence, all with regard to the *XSOL Mapping* software, and the provision of or failure to provide Support Services. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THE *XSOL Mapping* software.

ENTIRE AGREEMENT

This EULA (including any addendum or amendment to this EULA which is included with the *XSOL Mapping* software) are the entire agreement between you and **XSOL** relating to the *XSOL Mapping* and the support services (if any) and they supersede all prior or contemporaneous oral or written communications, proposals and representations with respect to the *XSOL Mapping* software or any other subject matter covered by this EULA. To the extent the terms of any **XSOL** policies or programs for support services conflict with the terms of this EULA, the terms of this EULA shall control.

7 Glossary

Advanced Query

An Advanced Query is a user's request for information when accessing a database. Advanced Queries enable you to create a selection of data that is used in other parts of the system, such as Lookup Lists and reports. This allows the selected subset of data to be searched when responding to a query, rather than searching the entire database. They are a useful (and for larger systems, mandatory) means of optimizing system performance.

Business Process Management

The mapping of the Processes and activities within a business, encompassing their design, analysis, optimization and monitoring of performance during execution.

Business System

The collection of Processes that are used to automate all aspects of a business.

Companion Process

Companion Processes are Processes that work in conjunction with their parent Process and can change information in existing Process instances. Companion Processes are used on-demand, rather than being scheduled as part of the Parent process, and are initiated from any model displaying information from the parent Process or from the To Do List for Stages to which a Companion Process is assigned. Any number of Companion Processes can be assigned to a single parent Process and each Companion Process can have any number of Initiation Rules assigned. Each Companion Process can optionally be flagged as usable on all completed Processes, all incomplete Processes, or both.

Control Group

A Table that usually comprises a code and description that is used to validate input content on other Tables and then enable grouping of data elements for viewing and/or reporting. These are typically validation- or grouping-type Tables.

Data Field

An individual unit of data held on a Table; the properties of each Data Field determine the type of data it can hold and the format in which it is entered, displayed and stored within the system.

Datafield Variable

Enables a datafield to be modifiable in a Worksheet.

Enterprise

The Enterprise is the top node of the hierarchy. An Enterprise is a container for the Business Processes of one or more Trading Entities. The Enterprise has the same capabilities as a Trading Entity, thus Roles, Reference Data, and Processes can all be defined.

Enterprise System Logic (ESL)

A model that describes business elements and activities in terms that can be applied to any Enterprise to enable the creation of automated Business Systems.

Escalation Level

Escalation Levels are a means of assigning Roles to any scheduled Stages in an *XSOL Automation* system after a set period of time. This moves overdue Stages onto the To Do Lists of managers or other employees and allows them to be actioned even if the first set of assigned users cannot access the system. Escalation Levels control which Roles or users can use specific Processes.

Expression

An Expression is a piece of business logic that can optionally include conditional statements the output of which is a result that is either stored in a Variable or queried by the Process Manager to determine the required future actions. For example, **Price * Quantity** represents the value of multiplying the Variables

Price and Quantity.

Flow node

See Task instance. Note that a Flow node is not the same as a Process Flow node (a Process Flow node can be any node in a Process Flow diagram, whereas a Flow node is only a Task assigned to a Stage).

Group

Groups are a means of listing items in relevant sections. They are only used to allow greater visibility for the designer and do not have any affect on a run-time system. Each area that uses Groups has one permanent Group called **(unassigned)**. This is used as a holder for all data items unless they were created in another Group. You can drag and drop items into and out of the **(unassigned)** Group as required. Unassigned items will still work effectively in the run-time system, since Groups do not affect the run-time system in any way.

Initiation Rule

Initiation Rules are used by Companion Processes to determine who can start a Companion Process. Companion Processes can be made available for all complete Processes, all incomplete Processes, or assigned to a Stage in the Process (useful if you wish to start a Companion Process in different Stages in the parent Process, and each Stage can be processed by different Roles, Teams, and/or users).

Lookup List

A Lookup List displays a selection of data from one or more Advanced Queries. Lookup Lists are used in Worksheets and allow you to present only the data that you want to a user at a particular point in a Process.

Macro Expression

A Macro Expression is an Expression or calculation, as defined above, which uses Parameters for the Variable elements (e.g. Quantity, Price) within the Expression or calculation. A Macro Expression is a define-once/use-many entity in that it can be assigned to many Process Variables. By using Macro Expressions any change applied to that Macro Expression are automatically applied to all Process Variables that the Macro Expression is assigned to.

Maintenance Task Worksheet

A form that allows a user to maintain table information in an XSOL system.

Model

A model is an XML (Extensible Markup Language) document based on ESL (Enterprise System Logic) terminology that defines the business model for a specific Enterprise in terms of its Trading Entities, Processes, Stages, Tasks, Reference Data, Controls and Data Fields. *XSOL Mapping* automatically loads the last edited model when the application is opened, if that option has been selected. Only one model can be edited by *XSOL Mapping* at one time.

Parameter

Parameters are used in Macro Expressions and are simply generic names for Variables used within the Expression. The Parameters are substituted by actual Process Variable names when the Macro Expression is assigned to a Process Variable, which will hold the result of the Macro Expression when it is run.

Parent Task

The Task from which Task instances are assigned to Stages.

Procedure Manual

A document that describes Business Processes and identifies activities by the Role that is responsible for each Task. It also contains Process Flow diagrams.

Process

The Business Processes (or day-to-day activities) by which a Trading Entity conducts its business whereby nothing is traded. Processes can be internal Processes, utilised to maintain Reference Data, adjust Aggregated data, or to undertake a Process in a defined and controlled fashion, or external Processes where external parties are involved but the item traded is typically dialogue (for example CRM).

A Process is a collection of Process Stages joined by workflow.

Process Flow

The sequence in which Stages are performed within a Process, or Tasks are performed within a Stage. Process Flow permits consecutive, concurrent, and conditional flow to be defined; once a Process has been initiated the XSOL Process Manager controls the what-happens-next scenario, using the Roles assigned to the Stages to achieve this.

Process Summary

Process Summaries provide a means of viewing Process instances in an *XSOL Automation* system. All Process Summaries display information in a spreadsheet format, to which you can add columns for more detailed information, and you can create read-only Worksheets that display information about a Process in a form. A Process Summary can optionally drill-down to Stage details, and Stage details can optionally drill-down to Task details.

Reference Data

An ESL (Enterprise System Logic) term that encompasses all table related data.

Reference Variable

Selects an instance of a Table. A Reference Variable is automatically created for a Table whenever a Table is created, and has the same name as the Table. Reference Variables contain Datafield Variables.

Relationship

A Table used to hold information that is relevant to one or more Resources and/or Control Groups that can also have optional range parameters specified. A Relationship has three separate segments in its UID: Tables; Ranges; and Datafields. A Relationship must contain a minimum of one Resource or Control Group but can contain more of either table type.

The range parameters are optional and include date, time, datetime (mutually exclusive with the date and time options), days of the week, and the mutually exclusive numeric and percentage breakpoints.

Resource

The discrete things that an Enterprise employs when undertaking its business, such as Products, Customers, Staff and Bank Accounts. They are typically tangible assets that would appear on the balance sheet.

Resource Extension

A logical group of Data Fields that is implicitly linked to a Resource but held as a separate Table. Each Resource may have many Resource Extensions and each Resource Extension may have a 1-to-1 or 1-to-many relationship to its parent Resource.

Role

The class of user to whom a Stage is assigned who is normally associated with a functional business area; for example, either broad - Accounting - or narrow - Credit Controller. Roles can be assigned to the Enterprise or Trading Entities.

Roles are items that can be reused within a model. Each time you create a Role within a Process you are creating a parent Role within the Enterprise or Trading Entity.

Stage

A Stage defines a step that comprises one or more Tasks, which can be scheduled within a Process.

Stored Variable

A permanent Variable that is stored in the event output from a Task. Stored Variables are used for data entry into Worksheets or to hold the result of a calculation or expression.

Tables

There are several classifications of Tables within *XSOL Mapping*: Resources; Resource Extensions; Control Groups; and Relationships. These classifications are utilized to enforce certain principles of ESL and to simplify model construction. A Table is effectively the physical recording of logical groups of Data Fields

within a database.

Task

A Task is an activity undertaken within a Stage that is not scheduled but rather initiated when any previous Task within the Stage is completed. Various types of Tasks can be represented including worksheets, reports, launching external applications (Excel, Word, ERP (Enterprise Resource Planning) program, etc.), manual operations, document operations, updating of Reference Data, sending emails, and so on. Tasks can be reused throughout an Enterprise or Trading Entity. A Task assigned to a Stage is known as a Task instance and the Task it is assigned from is known as the parent Task.

Task instance

A Task that has been assigned to a Process. The Task it is assigned from is known as the parent Task.

Task Process Flow

Process Flow within a Stage. Task Process Flow cannot be scheduled.

Task Worksheet

A form that allows a user to enter or update information in an XSOL system. Task Worksheets are used in the execution of Tasks that require user input.

To Do List

Models created by the *XSOL Mapping* can be used to create a running system using *XSOL Automation*. Each Stage in a Process will appear on a user's To Do List at some time after the Process has been instigated. You can set what columns are displayed and in what order they are sorted, based on the Variables that are available for the Stage.

Trading Entity

A discrete entity within the Enterprise through which business is conducted; each Trading Entity will have its own Roles, Reference Data, Processes, and Arithmetic data aggregation defined.

User Note

Used to enter information relevant to an item. Various User Note types can be defined to capture different types of information, such as Numeric or Word. All models contain the Version User Note, and any number of other User Notes can be defined as required.

Validation

A step that ensures that a model is logically correct.

Value Chain

A series of Processes by which the Trading Entity adds value to its business - from source to sale; for example, Buy, Make and Sell.

Variables

Variables are containers for holding data that is input through a Process. For example, in the Expression **Price * Quantity** both Price and Quantity are Variables. Variables have their own data type, much like datafields, which dictates what sort of information can be entered into them. Variables are used in Worksheets to accept data input by a user or to store information resulting from a calculation or Expression.

Where Used

Identifies where an ESL (Enterprise System Logic) component is used in a model.

Work Variable

A temporary Variable that holds information for the life of the Task that is input by a user or the result of a calculation or Expression.

Worksheet

A form that allows a user to enter or maintain information in an XSOL system. There are two types of Worksheets, Task Worksheets and Maintenance Task Worksheets.

8 Appendix

Refer to any of the following sections for more information.

8.1 Python modifications



Automation Systems Design only

The **Expression Editor** uses the Python scripting engine to function. The following topics describe the custom changes *XSOL Mapping* has made to Python.

For more information about standard Python, refer to <http://docs.python.org/>.

8.1.1 Naming conventions



Automation Systems Design only

All XSOL Python additions use de-facto Python naming conventions.

- Module and package names are in all lowercase.
- Class names always begin with an uppercase character. An uppercase character is also used to delimit "words" in the class name. For example, `Date ()`, `DateTime ()`, `FixedPoint ()`, `Percent ()`, `Days ()`, `Months ()`.
- Method names always begin with a lower case character. An uppercase character is also used to delimit "words" in the method name. For example, `DateTime.format ()`, `hasValue ()`, `hasNoValue ()`.
- Property names always begin with an upper case character. An uppercase character is also used to delimit "words" in the property name. For example, `DateTime ().Month`, `Time ().Hour`, `Date ().DayOfWeek`.
- Constant names are always in upper case. An underscore character is used to delimit "words" in constant names. For example, `UTC_ZONE`, `CURRENT_ZONE`.

8.1.2 Basic types



Automation Systems Design only

All of XSOL's Variable types are mapped to the following (refer to page 128 for more information):

- **Arithmetic** - `FixedPoint ()`
- **Auto Increment** - `FixedPoint ()`
- **Binary** - `Binary ()`
- **Date** - `Date ()`
- **Datetime** - `DateTime ()`
- **Days of the Week** - `WeekDayRange ()`
- **Document** - `Document ()`
- **Email Address** - `EmailAddress ()`

- **Memo** - `str ()` (Python generic type)
- **Notational** - `str ()` (Python generic type)
- **Numeric** - `FixedPoint ()`
- **Percent** - `Percent ()`
- **Phone Number** - `PhoneNumber ()`
- **Printer** - `PrinterRef ()`
- **Scientific** - `float ()` (Python generic type)
- **Time** - `Time ()`
- **User** - `UserRef ()`

As is the norm with Python, these types are immutable - once you set the value, you cannot change it. You can of course, create another object with a different value and assign that to the same variable.

If a Variable is not yet assigned (for example, the user has not yet keyed anything into the input widget), it will have the value of `NotSpecified`. To test that a variable is `NotSpecified`, use `hasValue ()` or `hasNoValue ()`.

8.1.3 Non-standard Python syntax



Automation Systems Design only

XSOL has had to modify Python's syntax slightly because Python does not directly support fixed point arithmetic.

- Any literal numbers that have a decimal place are converted to `FixedPoint`. For example, `123.45` gets converted to `FixedPoint ("123.45")`. Refer to page 216 for more information.
- To use Python's native float type, add `f` after the number. For example, `123.45f` gets converted to `float (123.45)`. Refer to page 217 for more information.
- To create a percent, add `p` after the number. For example, `123.45p` gets converted to `Percent ("123.45")`. Refer to page 219 for more information.
- If you use a single `=` character inside an `if` statement, *XSOL Mapping* converts that to `==`. For example, `if a = b:` gets converted to `if a == b:`. This conversion only takes place inside an `if` statement and as long as the `if` statement is on one line.

8.1.4 Reserved keywords



Automation Systems Design only

There are a number of reserved words that should not be used as names for Variables, Tables, Macro Expressions, Document Locations and Logical Printers because they might clash with names of functions and classes in Python. The following list includes XSOL-specific and Python-standard reserved keywords:

- `and`
- `assert`
- `average`

- Boolean
- break
- class
- coalesce
- continue
- count
- countBlank
- Date
- DateTime
- def
- del
- Document
- elif
- else
- EmailAddress
- except
- exec
- finally
- for
- from
- global
- hasValue
- hasNoValue
- if
- import
- in
- is
- lambda
- max
- min
- None
- not
- or

- pass
- PhoneNumber
- print
- raise
- return
- sum
- Time
- toString
- try
- WeekDay
- while
- Xml
- XSOL.P (as a prefix)
- yield

8.1.5 Global functions



Automation Systems Design only

The functions in this section are generally available for all data types used in *XSOL Mapping*.

average ()



Automation Systems Design only

average (values)

The **average ()** function returns the average of its non-null parameter values. The function ignores null parameter values.

Examples

```
g1 = average (1, 2, 3, 4)    #Returns 2.5
```

coalesce ()



Automation Systems Design only

coalesce (values)

The **coalesce ()** function returns the first non-null of its parameter values, reading from left to right. This is similar to the SQL (Structured Query Language) coalesce function.

Examples

```
#Returns the date of today if no date is available in the first two variables
g1 = coalesce (Date_Variable_1, Date_Variable_2, Date())
```

count ()

Automation Systems Design only

count (values)

The **count ()** function returns the number of non-null parameter values. The function ignores null parameter values.

Examples

```
g1 = count (2, 4, None, 5) #Returns 3
```

countBlank ()

Automation Systems Design only

countBlank (values)

The **count ()** function returns the number of null parameter values.

Examples

```
g1 = countBlank (2, 4, None, 5) #Returns 1
```

hasNoValue ()

Automation Systems Design only

hasNoValue (value)

The **hasNoValue ()** function returns the True if the parameter does not have a valid value.

Examples

```
# Return 0 if the Cost Variable has no value
if hasNoValue (Cost):
    return 0
```

hasValue ()

Automation Systems Design only

hasValue (value)

The **hasValue ()** function returns the True if the parameter does have a valid value.

Examples

```
# Return Cost if the Cost Variable has a valid value
if hasValue (Cost):
    return Cost
```

max ()

Automation Systems Design only

max (values)

The **max ()** function returns the largest parameter value.

Examples

```
g1 = max (8, 6, 11, 3)  #Returns 11
```

min ()



Automation Systems Design only

max (values)

The **min ()** function returns the smallest non-null parameter value. The function ignores null parameter values.

Examples

```
g1 = max (8, 6, 11, 3)  #Returns 3
```

sum ()



Automation Systems Design only

sum (values)

The **sum ()** function returns the total of all non-null parameter values. The function ignores null parameter values.

Examples

```
g1 = sum (8, 6, 11, 3)  #Returns 28
```

toString ()



Automation Systems Design only

toString (values)

The **toString ()** function returns a string that has concatenated all non-null parameters together (from left to right). The function ignores null parameter values.

Examples

```
# Returns a string suitable for use as Stage Summary
summary = toString ("Order number: ", Order_Number_Variable)
```

8.1.6 Binary operations



Automation Systems Design only

The **Binary ()** class returns a logical True or False. It is used to refer to a Binary data type when a Binary Variable is used in an Expression. It is mapped to XSOL's Binary type.

Examples

```
b1 = Binary (True)
b2 = int (b1)      #Returns 1
```

8.1.7 Date/Time operations



Automation Systems Design only

The following date and time functions are available:

- `Date ()` - Returns today's date at midnight local time displayed in the format `Day mm/dd/yy`.
- `DateTime ()` - Returns today's date and time displayed in the format `Day mm/dd/yy hh:mm:ss`.
- `Time ()` - Returns the current time for the local time zone displayed in the format `hh:mm:ss`.

All date and time calculations are accurate to one second, even if the `Date ()` class is used. When a `Date ()` value is returned the time component is removed. Similarly, when a `Time ()` value is returned the date component is removed.

Date and time types support the `+` and `-` operators, so it is possible to add or subtract date and time parts to or from a date or time. For example, you can add a `Time ()` object to a `DateTime ()` object.



Adding two date values may not achieve the results you expect, as dates are stored as the number of days relative to a datum.

The available date/time parts are:

- `Months ()`
- `Weeks ()`
- `Days ()`
- `Hours ()`
- `Minutes ()`
- `Seconds ()`

`Date ()`



Automation Systems Design only

The `Date ()` class returns today's date at midnight local time displayed in the format `Day mm/dd/yy`. It is mapped to XSOL's Date type.

The following syntax forms are available:

- `Date (Zone)` - Returns today's date. The time part is set to midnight local time. The date/time is then converted to GMT (Greenwich Mean Time) or the specified time zone.

Optional `Zone` argument:

- `CURRENT_ZONE` - (default) Returns the current date and time of the specified time zone. If the time zone is an integer, then the zone is in multiples of hours. If the zone is a tuple - `(hours, minutes)` - then the zone is specified in hours and minutes offset from GMT.
- `UTC_ZONE` - Returns UTC (Universal Time Clock) date and time.
- `Date ((year, month, day))` - Returns the date specified by the year, month, day values.
- `Date (DateTime)` - Returns the date portion from the specified `DateTime ()` object.

- `Date (Date)` - Copies the specified `Date ()` object.

The following properties and methods are available:

- `.Day` - Returns the day of the month.
- `.DayOfWeek` - Returns the day of the week (0 = Monday, 6 = Sunday).
- `.DayOfYear` - Returns the day of the year.
- `.Month` - Returns the month (1 = January, 12 = December).
- `.Year` - Returns the year.
- `.format ('string>')` - Returns the date and/or time formatted as a string. Refer to `time.strftime` in the Python documentation for more information.
- `.IsLeapYear ()` - Returns `True` if the year is a leap year and `False` if the year is not a leap year.

Examples

#These examples assume that today is Tuesday the 9th of August, 2005.
#The time is 1:05pm +12 UTC.

#Syntax examples

```
d1 = Date ()           #Returns Tue 08/09/05
d2 = Date (Zone = 5)   #Returns Tue 08/09/05
d3 = Date (Zone = (2,0)) #Returns Mon 08/08/05
d4 = Date (Zone = UTC_ZONE) #Returns Tue 08/09/05
d5 = Date ((1974,11,23)) #Returns Sat 11/23/74
dt = DateTime ()
d6 = Date (dt)         #Returns Tue 08/09/05
d7 = Date (d1)         #Returns Tue 08/09/05
```

#Property examples

```
dp1 = Date ().Day      #Returns 9
dp2 = Date ().DayOfWeek #Returns 1
dp3 = Date ().DayOfYear #Returns 220
dp4 = Date ().Month    #Returns 8
dp5 = Date ().Year     #Returns 2005
dp6 = Date ().format ('%A, %d %B, %Y') #Returns Tuesday, 09 August, 2005
dp7 = Date ().IsLeapYear () #Returns False
```

DateTime ()



Automation Systems Design only

The `Date ()` class returns today's date and time displayed in the format `Day mm/dd/yy hh:mm:ss`. It is mapped to XSOL's `Datetime` type.

The following syntax forms are available:

- `DateTime (Zone)` - Returns today's date and time. The date/time is then converted to GMT (Greenwich Mean Time) or the specified time zone.

Optional `Zone` argument:

- `CURRENT_ZONE` - (default) Returns the current date and time of the specified time zone. If the time zone is an integer, then the zone is in multiples of hours. If the zone is a tuple - (`hours`, `minutes`) - then the zone is specified in hours and minutes offset from GMT.
- `UTC_ZONE` - Returns UTC (Universal Time Clock) date and time.
- `DateTime ((year, month, day, hour, minute, second))` - Returns the date specified by the year, month, day, hour, minute, second values. All values default to now if not specified.
- `DateTime (DateTime)` - Copies the specified `DateTime ()` object.
- `DateTime (Date)` - Returns the date from the specified `Date ()` object.
- `DateTime (Time)` - Returns the time from the specified `Time ()` object.
- `DateTime (Date, Time)` - Returns the date and time from the specified `Date ()` and `Time ()` objects.

The following properties and methods are available:

- `.Day` - Returns the day of the month.
- `.DayOfWeek` - Returns the day of the week (0 = Monday, 6 = Sunday).
- `.DayOfYear` - Returns the day of the year.
- `.Month` - Returns the month (1 = January, 12 = December).
- `.Year` - Returns the year.
- `.Hour` - Returns the hour in 24-hour format.
- `.Minute` - Returns the minutes.
- `.Second` - Returns the seconds.
- `.format ('string>')` - Returns the date and/or time formatted as a string. Refer to `time.strftime` in the Python documentation for more information.
- `.IsLeapYear ()` - Returns `True` if the year is a leap year and `False` if the year is not a leap year.
- `.ConvertToUTCZone ()` - Converts the local date and time to UTC (Coordinated Universal Time) date and time. This is required when converting a `Date` type to a `Datetime` type.

Examples

#These examples assume that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.

```
#Syntax examples
dt1 = DateTime ()                #Returns Tue 08/09/05 12:05:28
dt2 = DateTime (Zone = 5)       #Returns Tue 08/09/05 5:05:28
dt3 = DateTime (Zone = (2,0))   #Returns Tue 08/09/05 2:05:28
dt4 = DateTime (Zone = UTC_ZONE) #Returns Tue 08/09/05 12:05:28
dt5 = DateTime ((1974, 11, 23, 12, 5, 28)) #Sat 11/23/74 12:05:28
dt6 = DateTime (dt1)            #Returns Tue 08/09/05 12:05:28
d1 = Date () + Days (1)         #Returns Wed 08/10/05 00:00:00
dt7 = DateTime (d1)              #Returns Wed 08/10/05 00:00:00
t1 = Time () - Hours (6)        #Returns Tue 08/09/05 06:05:28
dt8 = DateTime (t1)
```

```

dt9 = DateTime (d1, t1)                                #Returns Tue 08/09/05 12:05:28

#Property examples
dtp1 = DateTime ().Day                                  #Returns 9
dtp2 = DateTime ().DayOfWeek                            #Returns 1
dtp3 = DateTime ().DayOfYear                            #Returns 220
dtp4 = DateTime ().Month                                #Returns 8
dtp5 = DateTime ().Year                                 #Returns 2005
dtp6 = DateTime ().format ('%A, %d %B, %Y')            #Returns Tuesday, 09 August, 2005
dtp7 = DateTime ().IsLeapYear ()                        #Returns False
d2 = Date()
dtp8 = DateTime (d2).ConvertToUTCZone()                 #Returns the local zone date and time converted to UTC da

```

Days ()



Automation Systems Design only

Days (nnn)

The Days () class returns the number of days entered. This is used to add or subtract time periods.

Examples

```

#This example assumes that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.

```

```

time = DateTime () + Days (1)    #Returns Wed 08/10/05 12:05:28

```

DaysDiff ()



Automation Systems Design only

DaysDiff (From, To)

The DaysDiff () function returns the difference (positive or negative) in days between two date and time periods.

Examples

```

#This example assumes that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.

```

```

time1 = DateTime ()
time2 = DateTime () + Days (2)
diff = DaysDiff (time1, time2)    #Returns 2.0
return diff

```

Hours ()



Automation Systems Design only

Hours (nnn)

The Hours () class returns the number of hours entered. This is used to add or subtract time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.  
#The time is 12:05:28pm +12 UTC.
```

```
time = DateTime () + Hours (6)      #Returns Tue 08/09/05 18:05:28
```

HoursDiff ()



Automation Systems Design only

HoursDiff (From, To)

The HoursDiff () function returns the difference (positive or negative) in hours between two date and time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.  
#The time is 12:05:28pm +12 UTC.
```

```
time1 = DateTime ()  
time2 = DateTime () + Days (2)  
diff = HoursDiff (time1, time2)      #Returns 48.0  
return diff
```

Minutes ()



Automation Systems Design only

Minutes (nnn)

The Minutes () class returns the number of minutes entered. This is used to add or subtract time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.  
#The time is 12:05:28pm +12 UTC.
```

```
time = DateTime () - Minutes (25)    #Returns Tue 08/09/05 11:40:28
```

MinutesDiff ()



Automation Systems Design only

MinutesDiff (From, To)

The MinutesDiff () function returns the difference (positive or negative) in minutes between two date and time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.  
#The time is 12:05:28pm +12 UTC.
```

```
time1 = DateTime ()  
time2 = DateTime () + Days (2)  
diff = MinutesDiff (time1, time2)    #Returns 2880.0  
return diff
```


Months ()



Automation Systems Design only

Months (nnn)

The **Months ()** class returns the number of months entered. This is used to add or subtract time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.
```

```
time = DateTime () - Months (2)      #Returns Thu 06/09/05 12:05:28
```

Seconds ()



Automation Systems Design only

Seconds (nnn)

The **Seconds ()** class returns the number of seconds entered. This is used to add or subtract time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.
```

```
time = DateTime () + Seconds (30)    #Returns Tue 08/09/05 12:05:58
```

SecondsDiff ()



Automation Systems Design only

SecondsDiff (From, To)

The **SecondsDiff ()** function returns the difference (positive or negative) in seconds between two date and time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.
```

```
time1 = DateTime ()
time2 = DateTime () + Days (2)
diff = SecondsDiff (time1, time2)    #Returns 172800
return diff
```

Time ()



Automation Systems Design only

The **Time ()** class returns the current time for the local time zone displayed in the format **hh:mm:ss**. It is mapped to XSOL's Time type.

The following syntax forms are available:

- `Time (Zone)` - Returns the current time for the local time zone.
- `Time ((year, month, day, hour, minute, second))` - Returns the time specified by the year, month, day, hour, minute, second values.
- `Time (DateTime)` - Returns the time portion from the specified `DateTime ()` object.
- `Time (Time)` - Copies the specified `Time ()` object.

The following properties and methods are available:

- `.Hour` - Returns the hour in 24-hour format.
- `.Minute` - Returns the minutes.
- `.Second` - Returns the seconds.
- `.format ('string>')` - Returns the date and/or time formatted as a string. Refer to `time.strftime` in the Python documentation for more information.

Examples

#These examples assume that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.

```
#Syntax examples
t1 = Time ()           #Returns 12:05:28
t2 = Time ((12,5,28))  #Returns 12:05:28
dt = DateTime ()
t3 = Time (dt)         #Returns 12:05:28
t4 = Time (t1)         #Returns 12:05:28

#Property examples
tp1 = Time ().Hour     #Returns 12
tp2 = Time ().Minute   #Returns 5
tp3 = Time ().Second   #Returns 28
tp4 = Time ().format ('%H:%M') #Returns 12:05
```

WeekDay ()



Automation Systems Design only

The `WeekDay ()` class returns the name of the given week day in the format Monday, Tuesday, Wednesday, etc.

The following syntax forms are available:

- `WeekDay ()` - Returns today's day of the week.
- `WeekDay (DateTime)` - Returns the day of the week for the specified `DateTime ()` object.
- `WeekDay (WeekDay)` - Copies the specified `WeekDay ()` object.
- `WeekDay (nnn)` - Returns the day of the week using a number from 1 to 7 (1 = Sunday, 7 = Saturday).
- `WeekDay (s)` - Returns the day of the week using a string. The available string values are: 'Su', 'Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa'. The string values are case-insensitive.

Examples

#This example assumes that today is Tuesday the 9th of August, 2005.
 #The time is 12:05:28pm +12 UTC.

```
wd1 = WeekDay ()           #Returns Tuesday
dt = DateTime() - Days (1)
wd2 = WeekDay (dt)         #Returns Monday
wd3 = WeekDay (wd1)        #Returns Tuesday
wd4 = WeekDay (2)          #Returns Monday
wd5 = WeekDay ('we')       #Returns Wednesday
wd6 = str(wd5)             #Returns Wednesday
```

WeekDayRange ()



Automation Systems Design only

The `WeekDayRange ()` class returns a set of days of the week. This is useful whenever a range of week days are required, such as setting the working week of a retail store as being Monday to Friday. It is mapped to XSOL's Days of the Week type.

The following syntax forms are available:

- `WeekDayRange ()` - Returns a set of days that only includes today's day of the week.
- `WeekDayRange (DateTime)` - Returns a set of days that only includes the day of the week for the specified `DateTime ()` object.
- `WeekDayRange (WeekDay)` - Copies the specified `WeekDayRange ()` object.
- `WeekDayRange (list)` - Returns a set of days that you can specify in a number of ways:
 - `[n, n, n]` - List that corresponds to days of the week using numbers from 1 to 7 (1 = Sunday, 7 = Saturday).
 - `s` - List that corresponds to days of the week using a string. The available string values are: 'Su', 'Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa'. The string values are case-insensitive.
 - `(CONSTANT)` - List that corresponds to days of the week using constants. The available constants are: SUNDAY, MONDAY, TUESDAY, etc.
- `WeekDayRange (s)` - Returns the day of the week using a string. The available string values are: 'Su', 'Mo', 'Tu', 'We', 'Th', 'Fr', 'Sa'. The string values are case-insensitive.

Examples

#This example assumes that today is Tuesday the 9th of August, 2005.
 #The time is 12:05:28pm +12 UTC.

```
wdr1 = WeekDayRange ()           #Returns Tuesday
dt = DateTime() - Days (1)
wdr2 = WeekDayRange (dt)         #Returns Monday
wdr3 = WeekDayRange (wdr1)       #Returns Tuesday
wdr4 = WeekDayRange ([2,3,4,5,6]) #Returns Monday, Tuesday, Wednesday, Thursday, Friday
wdr5 = WeekDayRange ('Sa Su')    #Returns Saturday, Sunday
wdr6 = len (wdr5)               #Returns 2
wdr7 = WeekDayRange ((SUNDAY, MONDAY)) #Returns Sunday, Monday
```

Weeks ()



Automation Systems Design only

The `Weeks ()` class returns the number of weeks entered. This is used to add or subtract time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.
```

```
time = DateTime () + Weeks (2)    #Returns Tue 23/09/05 12:05:58
```

WeeksDiff ()



Automation Systems Design only

`WeeksDiff (From, To)`

The `WeeksDiff ()` function returns the difference (positive or negative) in weeks between two date and time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.
```

```
time1 = DateTime ()
time2 = DateTime () + Days (7)
diff = WeeksDiff (time1, time2)    #Returns 1.0
```

Years ()



Automation Systems Design only

`Years (nnn)`

The `Years ()` class returns the number of years entered. This is used to add or subtract time periods.

Examples

```
#This example assumes that today is Tuesday the 9th of August, 2005.
#The time is 12:05:28pm +12 UTC.
```

```
time = DateTime () + Years (1)    #Returns Wed 08/09/06 12:05:58
```

8.1.8 Document operations



Automation Systems Design only

The functions in this section are all used to manipulate documents and document locations.

Document ()



Automation Systems Design only

`Document (location, relative_path)`

The `Document ()` class creates a Document type using a document location and relative path. It is mapped to XSOL's Document type.

The following properties are available:

- `.Location` - Returns the document location used by the document.
- `.RelativePath` - Returns the relative path used by the document.
- `.getFileURL ()` - Returns the URL of the file in the document location. Only available in Script Tasks - refer to page 221 for more information. When certfying the Expression this uses the **Root document location folder** option - refer to page 169 for more information.
- `.getLocalFileName ()` - Creates a copy of the document in the document location and returns the filename and location of the copy. Only available in Script Tasks - refer to page 221 for more information. When certfying the Expression this uses the **Root document location folder** option - refer to page 169 for more information.

Examples

```
#Templates is an existing Document Location
d1 = Document (Templates, '/accounts/invoice.dot') #Returns [XSOL_URL]/accounts/invoice.dot
d2 = d1.Location                                #Returns Document Location
d3 = d1.RelativePath                            #Returns /accounts/invoice.dot
```

DocumentLocation ()



Automation Systems Design only

The `DocumentLocation ()` class returns a Document Location. It is used to refer to a Document Location whenever a Document Location is used in an Expression.



Do not use the `DocumentLocation ()` class to create a Document Location reference in an Expression.

8.1.9 Email operations



Automation Systems Design only

The `EmailAddress ()` class returns a string as an email address. It is mapped to XSOL's Email Address type.

The following syntax form is available:

- `EmailAddress (s)` - Returns the string as an email address. The string needs to be entered in a valid email address format.

The following properties and methods are available:

- `.Name` - Returns the display name portion of the email address.
- `.Server` - Returns the domain name portion of the email address.
- `.User` - Returns the user name portion of the email address.

Examples

```

e1 = EmailAddress ('John Smith<johns@acme.net>') #Returns John Smith<johns@acme.net>
e2 = e1.Name #Returns John Smith
e3 = e1.Server #Returns acme.net
e4 = e1.User #Returns johns

```

8.1.10 File operations



Automation Systems Design only

The functions in this section are all used to manipulate files and file names. The `FileName ()` class is the containing class for the other functions.

appendDirectory ()



Automation Systems Design only

`appendDirectory (s)`

The `appendDirectory ()` function adds a directory to the end of a path.

Examples

```

fn1 = FileName ('C:\\dev\\')
fn1.appendDirectory ('temp') #Changes fn1 to C:\\dev\\temp

```

deleteFile ()



Automation Systems Design only

`deleteFile ()`

The `deleteFile ()` function deletes the file. It returns 1 if the file has been successfully deleted, or 0 if the file has not been deleted or does not exist.

Examples

```

fn1 = FileName ('C:\\dev\\to_do.txt')
fn2 = fn1.deleteFile () #Returns 1 if the file is deleted, or 0 if it is not

```

exists ()



Automation Systems Design only

`exists ()`

The `exists ()` function returns True if the file exists, or False if it does not exist.

Examples

```

fn1 = FileName ('C:\\dev\\to_do.txt')
fn2 = fn1.exists () #Returns True if the file exists, or False if does not

```

FileName ()



Automation Systems Design only

The `FileName ()` class encapsulates a file name. `FileName ()` accepts slash `/` and backslash `\` characters but always converts slashes to backslashes. All backslash characters must be entered twice to avoid being used as a special character.

The following syntax forms are available:

- `FileName (s)` - Returns a filename using a given string parameter.
- `FileName (FileName)` - Copies the specified `FileName ()` object.

The following properties are available:

- `.Delimiter` - Returns the character used as the path delimiter.
- `.Ext` - Returns the extension part of the filename.
- `.Filename` - Returns the name and extension part of the filename.
- `.Filepath` - Returns the full filename.
- `.Name` - Returns the name part of the filename, excluding extension.
- `.Path` - Returns the path part of the filename, including final backslash.

Examples

```
fn1 = FileName ('C:\\dev\\to_do.txt') #Returns C:\dev\to_do.txt
fn2 = FileName (fn1)                #Returns C:\dev\to_do.txt
fn3 = fn1.Delimiter                  #Returns \
fn4 = fn1.Ext                        #Returns txt
fn5 = fn1.Filename                   #Returns to_do.txt
fn6 = fn1.Filepath                   #Returns C:\dev\to_do.txt
fn7 = fn1.Name                       #Returns to_do
fn8 = fn1.Path                       #Returns C:\dev\
```

forceDirectories ()



Automation Systems Design only

`forceDirectories ()`

The `forceDirectories ()` function ensures that a path is correct by creating any missing directories.

Examples

```
fn1 = FileName ('C:\\dev\\temp\\to_do.txt')
fn2 = fn1.forceDirectories () #Creates dev and/or temp directories, as required.
```

move ()



Automation Systems Design only

`move (s)`

The `move ()` function renames the file or moves the file to the specified directory.

Examples

```
fn1 = FileName ('C:\\dev\\to_do.txt')
fn1.move (FileName ('C:\\dev\\to_do.doc')) #Renames to_do.txt to to_do.doc
```

pathList ()

Automation Systems Design only

pathList ()

The `pathList ()` function returns the path as a list. The following rules are followed:

- If a drive is present, it is the first item.
- If the path is a UNC path, the first item is a backslash.
- If the path is from root, the first item is a backslash.
- Backslashes are stripped from all path items.

Examples

```
fn1 = FileName ('C:\\dev\\temp\\to_do.txt')
fn2 = fn1.pathList () #Returns (C:, dev, temp, )
```

removeDirectories ()

Automation Systems Design only

removeDirectories ()

The `removeDirectories ()` function deletes the last directory in the path and all sub-directories, if any.

Examples

```
fn1 = FileName ('C:\\dev1\\temp\\to_do.txt')
fn2 = fn1.removeDirectories () #Deletes the temp directory and all sub-directories
```

8.1.11 Notational operations

Automation Systems Design only

str (object)

The `str ()` class is a Python generic type that stores characters. It is mapped to XSOL's Notational and Memo types.

Examples

```
fp1 = FixedPoint ('123.45')
s1 = str (fp1) #Returns 123, which is now a string
```

8.1.12 Number operations

Automation Systems Design only

The `FixedPoint ()` and `Percent ()` types support the following operators:

- + - Addition
- - - Subtraction

- `/` - Division
- `*` - Multiplication
- `**` - Power
- `mod` - Modulus
- `//` - Integer division
- `divmod` - Integer division and returns remainder (not supported by `Percent` ())

Refer to page 218 for more information on how numbers and percentiles can be used together.

All calculations use 9 decimal digit precision. When a `FixedPoint` () or `Percent` () value is returned, it is rounded to the precision of the variable that will receive the result. `str (FixedPoint ())` or `str (Percent ())` always returns a number in string format that does not have trailing decimal zeros. For example:

- `fp = str (FixedPoint ("123.45000"))` returns 123.45
- `fp = str (FixedPoint ("123.00000"))` returns 123
- `fp = str (FixedPoint ("12345000"))` returns 12345000

FixedPoint ()



Automation Systems Design only

The `FixedPoint` () class returns a number with decimals. Any number with a decimal in an Expression will be converted to `FixedPoint` (). The `FixedPoint` () type supports the basic arithmetic and comparison operators. It is mapped to XSOL's Arithmetic, Auto Increment, and Numeric types.

The following syntax form is available:

- `FixedPoint (nnn, nnn)` - Returns the number as a decimal. You can optionally enter the number of decimal places to round to before returning.

The following properties and methods are available:

- `.round` - Sets the type of rounding used. This is a global setting and applies to all rounding in all Expressions that occur in after the setting has been changed. The following rounding options are available:
 - `normalRounding` - (default) Rounds 0-4 downwards, and 5-9 upwards.
 - `bankersRounding` - Rounds 0-4 downwards, and 6-9 upwards. 5 is rounded downwards if the last digit before the decimal place is odd, or upwards if the last digit before the decimal place is even.
- `.frac` () - Returns the numbers after the decimal place only.
- `.toWords` () - Returns the numbers before the decimal place as words, and the numbers after the decimal place as a fraction.
- `.toWordsCents (nnn)` - Returns the numbers after the decimal place as words. You can optionally enter the number of decimal places to round to before returning.
- `.toWordsDollars` () - Returns the numbers before the decimal place as words.
- `.toZeroPaddedString (nnn)` - Returns the number and converts to a string of specified width. Zeros are added as required to the left of the number.

- `.StrPrecision` - Sets the precision of the rounding applied to a number and converts the number to a string.
- `.precision` - Sets the precision of the rounding applied to a number.

Examples

```
fp1 = 123.45                                #Equivalent to FixedPoint ('123.45')
fp2 = FixedPoint ('123.45')                 #Returns 123.45
fp3 = FixedPoint ('123.45', 1)              #Returns 123.5 using normal rounding
FixedPoint.round = bankersRounding         #Sets bankers rounding
fp4 = FixedPoint ('123.45', 1)              #Returns 123.5 using bankers rounding
FixedPoint.round = normalRounding          #Sets normal rounding
fp5 = fp1.frac ()                           #Returns 0.45
fp6 = fp1.toWordsCents (2)                  #Returns Forty Five
fp7 = fp1.toWordsCents (3)                  #Returns Four Hundred Fifty
fp8 = fp1.toWordsDollars ()                 #Returns One Hundred Twenty Three
fp9 = fp1.toZeroPaddedString (6)           #Returns 000123
fp1.StrPrecision = 1                       #Returns 123.5
fp1.precision = 1                          #Returns 123.5
fp10 = fp1.toWords ()                      #Returns One Hundred Twenty Three 5/10
```

float ()



Automation Systems Design only

`float (object)`

The `float ()` class is a built-in Python type that supports floating point and complex numbers. The `float ()` type supports the basic arithmetic and comparison operators. Any arithmetic with a `FixedPoint ()` type will result in a `FixedPoint` type. It is mapped to XSOL's Scientific type.

The following syntax forms are available:

- `float (nnn)` - Returns the number as a floating point number.
- `f` - Added after the number to indicate that the number is a floating point number.

Examples

```
fp1 = FixedPoint ('123.45')
f1 = float (fp1)                            #Returns 123.45, which is now a floating point number
f2 = 123.45f                                #Returns 123.45, which is now a floating point number
```

int ()



Automation Systems Design only

`int (object)`

The `int ()` class is a built-in Python type that supports integers (numbers that do not have decimal points). The `int ()` type supports the basic arithmetic and comparison operators. It is not used in XSOL and is only required if you need to supply an integer value to an external application.

Examples

```
fp1 = FixedPoint ('123.45')
int1 = int (fp1)                            #Returns 123, which is now an integer
```

8.1.13 Percent operations



Automation Systems Design only

Addition

- **Percent + Percent:** Adds two percentage values and returns a result as a **Percent**. For example, `Percent ("1") + Percent ("2") = Percent ("3")`.
- **FixedPoint + Percent:** Adds a percentage to the **FixedPoint** number and returns a **FixedPoint**. For example, `FixedPoint ("100") + Percent ("10") = FixedPoint ("110")`. This covers a markup pricing operation.
- **Percent + FixedPoint:** Same as `FixedPoint + Percent`.
- **Number + Percent:** Number is one of `int`, `long`, or `float`. Number is first converted to a **FixedPoint** and then added to the **Percent**. For example, `10 + Percent ("5") = FixedPoint ("10.5")`.
- **Percent + Number:** Same as `Number + Percent`.

Subtraction

- **Percent - Percent:** Subtracts one percentage value from another and returns a result as a **Percent**. For example, `Percent ("5") - Percent ("1") = Percent ("4")`.
- **FixedPoint - Percent:** Subtracts a percentage from the **FixedPoint** number and returns a **FixedPoint**. For example, `FixedPoint ("100") - Percent ("10") = FixedPoint ("90")`. This covers a discount pricing operation.
- **Percent - FixedPoint:** Not supported.
- **Number - Percent:** Number is one of `int`, `long`, or `float`. Number is first converted to a **FixedPoint** and then **Percent** is subtracted. For example, `10 - Percent ("5") = FixedPoint ("9.5")`.
- **Percent - Number:** Not supported.

Multiplication

- **Percent * Percent:** Multiplies two percentages together and returns the result as a **Percent**. For example, `Percent ("5") * Percent ("10") = Percent ("0.5")`.
- **FixedPoint * Percent:** Multiplies the **FixedPoint** number by **Percent** and returns the result as a **FixedPoint**. For example, `FixedPoint ("200") * Percent ("5") = FixedPoint ("10")`. You would use this when you want to return the percentage of a number.
- **Percent * FixedPoint:** Same as `FixedPoint * Percent`.
- **Number * Percent:** Number is one of `int`, `long`, or `float`. Number is first converted to a **FixedPoint** and then multiplied to a **Percent**. For example, `220 * Percent ("5") = FixedPoint ("11")`.
- **Percent * Number:** Same as `Number * Percent`.

Division

- **Percent / Percent:** Divides one percent by another and returns the result as a **Percent**. For example, `Percent ("10") / Percent ("5") = Percent ("200")`.

- **FixedPoint / Percent**: The result is a **FixedPoint** number that if multiplied by **Percent** returns a **FixedPoint**. For example, `FixedPoint ("5") / Percent ("10") = FixedPoint ("50")`, because `FixedPoint ("50") * Percent ("10") = FixedPoint ("5")`. You would use this to work out a GST exclusive price from the GST value and rate.
- **Percent / FixedPoint**: Not supported.
- **Number / Percent**: **Number** is one of `int`, `long`, or `float`. **Number** is first converted to a **FixedPoint** and then divided by **Percent**. For example, `5 / Percent ("10") = FixedPoint ("50")`.
- **Percent / Number**: Not supported.

Percent ()



Automation Systems Design only

The **Percent ()** class returns a percentile number. The **Percent ()** type supports the basic arithmetic and comparison operators, with some limitations - refer to page 218 for more information. It is mapped to XSOL's Percent type.

The following syntax forms are available:

- **Percent (nnn)** - Returns the number as a percentile.
- **p** - Used instead of the `%` symbol to indicate that the number is a percent. `%` is used in standard Python to indicated modular division.

The following properties and methods are available:

- **.round** - Sets the type of rounding used. This is a global setting and applies to all rounding in all Expressions that occur in after the setting has been changed. The following rounding options are available:
 - **normalRounding** - (default) Rounds 0-4 downwards, and 5-9 upwards.
 - **bankersRounding** - Rounds 0-4 downwards, and 6-9 upwards. 5 is rounded downwards if the last digit before the decimal place is odd, or upwards if the last digit before the decimal place is even.
- **.frac ()** - Returns the numbers after the decimal place only.
- **.toWords ()** - Returns the numbers before the decimal place as words, and the numbers after the decimal place as a fraction.
- **.toWordsCents (nnn)** - Returns the numbers after the decimal place as words. You can optionally enter the number of decimal places to round to before returning.
- **.toWordsDollars ()** - Returns the numbers before the decimal place as words.
- **.toZeroPaddedString (nnn)** - Returns the number and converts to a string of specified width. Zeros are added as required to the left of the number.
- **.StrPrecision** - Sets the precision of the rounding applied to a number and converts the number to a string.
- **.precision** - Sets the precision of the rounding applied to a number.

Examples

```

p1 = Percent ('20.45')           #Returns 20.45%
p2 = 20.45p                      #Returns 20.45%
p3 = Percent ('20.45', 1)        #Returns 20.5 using normal rounding
Percent.round = bankersRounding  #Sets bankers rounding
p4 = Percent ('20.45', 1)        #Returns 20.4 using bankers rounding
Percent.round = normalRounding  #Sets normal rounding
p5 = p1.frac ()                  #Returns 0.2045
p6 = p1.toWordsCents (2)         #Returns Twenty
p7 = p1.toWordsCents (3)         #Returns Two Hundred Five
p8 = p1.toWordsDollars ()        #Returns Twenty
p9 = p1.toZeroPaddedString (2)   #Returns Zero
p1.StrPrecision = 1              #Returns 20.5%
p1.precision = 1                 #Returns 20.5%
p10 = p1.toWords ()              #Returns Twenty 5%/100

```

8.1.14 Phone operations



Automation Systems Design only

The `PhoneNumber ()` class returns a string as a phone number. It is mapped to XSOL's Phone Number type.

The following syntax form is available:

- `PhoneNumber (s)` - Returns the string as a phone number. The string needs to be entered in the format `+country_code area_code number extension`. The country code is preceded by a `+` and the extension by an `x` - both are optional.

The following properties and methods are available:

- `.Area` - Returns the area code portion of the phone number.
- `.Country` - Returns the country code portion of the phone number.
- `.Extension` - Returns the extension portion of the phone number.
- `.Number` - Returns the local number portion of the phone number.

Examples

```

pn1 = PhoneNumber ('+64 09 5246412 x648') #Returns +64-09-5246412 ext648
pn2 = pn1.Area                           #Returns 09
pn3 = pn1.Country                         #Returns 64
pn4 = pn1.Extension                       #Returns 648
pn5 = pn1.Number                         #Returns 5246412

```

8.1.15 Printer operations



Automation Systems Design only

The `PrinterRef ()` class returns a logical printer. It is used to refer to a Logical Printer whenever a Logical Printer is used in an Expression. It is mapped to XSOL's Printer type.



Do not use the `PrinterRef ()` class to create a Logical Printer reference in an Expression.

8.1.16 Reference Variable operations



Automation Systems Design only

The `InstanceRef ()` class returns a Reference Variable. It is used to refer to a Reference Variable whenever a Reference Variable is used in an Expression.



Do not use the `InstanceRef ()` class to create a Reference Variable reference in an Expression.

8.1.17 Script Task operations



Automation Systems Design only

The functions in this section can only be used with Script Tasks - refer to page 71 for more information.

AddFileToDocument ()



Automation Systems Design only

`AddFileToDocument (stream)`

The `AddFileToDocument ()` class allows a local file to be copied into a document location. Only available in Script Tasks. When certifying the Expression this uses the **Root document location folder** option - refer to page 169 for more information.

Examples

```
# Customer Documents is a document location. C:\a.txt must exist
Customer_Invoice = AddFileToDocument ('C:\\a.txt', Customer_Documents, '/C001/Inv001.doc')
print Customer_Invoice      #Copies the file to the document location.
```

getTaskDataXml ()



Automation Systems Design only

`getTaskDataXml ()`

The `getTaskDataXml ()` function returns all variables and system values that are assigned to the task as an XML document DOM (Document Object Model). This method is useful when interfacing with external systems that already talk XML. An XML document DOM can also be the result of the expression. This will then set the values of the variables. Only available in Script Tasks.

Syntax notes:

- All elements use the Variable/Datafield **External ID** property as the element name.
- Most Variables use `xsi:nil="true"` to indicate a null value.
- All reference Variables (user, table, or print references) use `instanceId="{00000000-0000-0000-0000-000000000000}"` to indicate a null value.
- Reference Variables are optimized to only return the `UID` data and other datafields used in some way by the Task. This means that to return datafields, the datafield must either be explicitly referenced in the script or used elsewhere in the Task.
- Non-global System Values appear within the Process data along with other Variable values but are in the `http://www.xsol.com/esl/systemValues` namespace.

- All global system values appear within the same named element and again have the same namespace URI.

When certifying in *XSOL Mapping*:

- Run-time values are not available.
- All elements with `instanceId` attributes will not have `definitionId` attributes.
- The `:taskVersion>` element is not available.
- The `>` element `instanceId` attribute is null.

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<xscr:scriptTaskData xmlns:esl="http://www.xsol.com/ESLSchema" xmlns:xscr="http://www.xsol.com/scriptTa
```

```
    <process instanceId="{73867C99-6050-4D92-9663-CC224148B111}" definitionId="{AB32AD6D-AADF-4669-B
```

```
        <xsv:taskVersion>1_0</xsv:taskVersion>
```

```
        <Arithmetic>15</Arithmetic>
```

```
        <Auto_Increment>1</Auto_Increment>
```

```
        <Binary_False>>false</Binary_False>
```

```
        <Binary_True>>true</Binary>
```

```
        <Date>2005-12-14</Date>
```

```
        <Date_Time>2005-12-13T22:40:36.623</Date_Time>
```

```
        <Days_Of_Week>Su Mo Tu We Th Fr Sa</Days_Of_Week>
```

```
        <Document>xds://{99183690-9281-4758-A9DB-E53C4D262EF9}/A.doc</Document>
```

```
        <Copied_Document>xds://{99183690-9281-4758-A9DB-E53C4D262EF9}/A.doc</Copied_Document>
```

```
        <Email_Address>david@acme.org.nz</Email_Address>
```

```
        <Memo>First line of memo
```

```
Second line of memo</Memo>
```

```
        <Notational>Hello</Notational>
```

```
        <Numeric_1>1.00</Numeric_1>
```

```
        <Numeric_2>0</Numeric_2>
```

```
        <Numeric_3>-5</Numeric_3>
```

```
        <Numeric_Null xsi:nil="true" />
```

```
        <Percent>5</Percent>
```

```

    <Phone_Number>+64 9 5147832</Phone_Number>

    <Printer instanceId="{F48B5559-D7FA-4E8A-96F7-DAFC56199IJ5}">

        <description xsi:nil="false">HP LaserJet 5</description>

    </Printer>

    <Role instanceId="{C48B8889-D7FA-4E8A-96F7-DAFC56199EF8}">

        <description xsi:nil="false">All Staff</description>

    </Role>

    <Scientific>343</Scientific>

    <Time>22:40:36.623</Time>

    <User instanceId="{B48B2229-D7FA-4E8A-96F7-DAFC56199AF0}">

        <userid xsi:nil="false">administrator</userid>

        <firstname xsi:nil="false">System</firstname>

        <lastname xsi:nil="false">Administrator</lastname>

        <emailAddress xsi:nil="true" />

        <description xsi:nil="false">Default System Administrator</description>

    </User>

    <Customer instanceId="{D42BC9E9-7252-48CE-B32C-6DBBFC42B79F}" definitionId="{739A5FAC-65C8-D...}>

        <Code>123456</Code>

        <Name>Jimmy</Name>

        <Type instanceId="{754A18A0-4908-4392-8090-351CE87E55DF}" definitionId="{7009418C-D...}>

            <Code>B</Code>

            <Description>Type B</Description>

        </Type>

    </Customer>

</process>

<xsv:globalSystemValues>

    <xsv:currentUserReference instanceId="{B48B2229-D7FA-4E8A-96F7-DAFC56199AF0}">

```



```

        <userid>administrator</userid>

        <firstname>System</firstname>

        <lastname>Administrator</lastname>

        <emailAddress xsi:nil="true" />

        <description>Default System Administrator</description>

    </xsv:currentUserReference>

    <xsv:enterpriseCaption>Enterprise</xsv:enterpriseCaption>

    <xsv:modelId>uuid_73B44A07-E723-44C7-ABDD-BC39E6839FB8</xsv:modelId>

    <xsv:modelVersion>1</xsv:modelVersion>

    <xsv:modelCommittedAt>2005-12-13T22:40:36.623</xsv:modelCommittedAt>

    <xsv:tradingEntityCaption>ACME Ltd</xsv:tradingEntityCaption>

    <xsv:tradingEntityVersion>1_0</xsv:tradingEntityVersion>

    <xsv:firstDayOfWeek>7</xsv:firstDayOfWeek>

</xsv:globalSystemValues>

</xscr:scriptTaskData>

```

8.1.18 Stream operations



Automation Systems Design only

The `Stream ()` class and sub-classes allow *XSOL Automation* to directly manipulate file, memory, and string streams. This normally is only required for interfacing with third-party applications.

FileStream ()



Automation Systems Design only

`FileStream (Filename, mode, sharemode)`

The `FileStream ()` class allows *XSOL Automation* to directly manipulate a file stream. Several parameters are used to create a `FileStream ()` instance:

- **Filename** - The file that will be created, read, or written to.
- **mode** - How the file is accessed. The following values are available:
 - **fmCreate** - Creates the file. If the file already exists and is not right-protected, the file is deleted and re-created.
 - **fmOpenRead** - Opens the file as a read-only file. Returns an error if the file does not exist.

- `fmOpenWrite` - Opens the file as a writeable file. Returns an error if the file does not exist.
- `fmOpenReadWrite` - Opens the file for both reading and writing operations. Returns an error if the file does not exist.
- `sharemode` - Determines how the stream is used by other processes. The following values are available:
 - `fmShareCompat` - (default) All processes who are also using `fmShareCompat` can access this stream. This setting handles the DOS method of accessing files.
 - `fmShareExclusive` - No other processes can access this stream.
 - `fmShareDenyWrite` - Other processes can not write to this stream.
 - `fmShareDenyRead` - Other processes can not read from this stream.
 - `fmShareDenyNone` - All processes can access this stream.

The following properties and methods are available:

- `.close ()` - Closes the file stream.
- `.copyFrom (stream)` - Copies the specified stream and returns the number of bytes copied.
- `.readFloat ()` - Returns a floating point number from the stream.
- `.readInt ()` - Returns a integer number from the stream.
- `.readLong ()` - Returns a long number from the stream. Long is a native Python data type, refer to <http://docs.python.org/> for more information.
- `.readStr (n)` - Returns a string from the stream based on the number of bytes to be read.
- `.seek (offset, whence)` - Moves the stream pointer position. Offset is the number of bytes to move, and whence specifies where to start moving from (0 = beginning of the stream, 1 = the current pointer location in the stream, 2 = the end of the stream).
- `.write (arg)` - Write to the stream. You can write to the stream using any of Python's native data types.
- `.Position` - Return the pointer position in the stream.
- `.Size` - Returns the byte size of the stream.

Examples

```
#Create a text file
fs1 = FileStream ("C:\\temp\\ticket.txt", fmCreate, fmShareCompat)

fs2 = fs1.copyFrom (fs1)    #Returns 0, as the file has just been created
fs3 = fs1.readFloat ()      #Returns 2
fs4 = fs1.readInt ()        #Returns 1
fs5 = fs1.readLong ()       #Returns 2
fs6 = fs1.readStr (10)      #Returns binary text, as the file has just been created
fs7 = fs1.seek (10, 0)      #Returns 10, as the pointer is now at the 10th byte
fs8 = fs1.Position          #Returns 0, as the file has just been created
fs9 = fs1.Size              #Returns 0, as the file has just been created

try:
    #Write file details
    fs1.write ("Travel Details:\n")
```

```

fs1.write ("Air Acme 238, depart 11:20, arrive 16:30.\n")

finally:
    # Close the file
    fs1.close ()

```

MemoryStream ()



Automation Systems Design only

MemoryStream (stream)

The **MemoryStream ()** class allows *XSOL Automation* to directly manipulate a memory stream.

The following properties and methods are available:

- **.copyFrom (stream)** - Copies the specified stream and returns the number of bytes copied.
- **.readFloat ()** - Returns a floating point number from the stream.
- **.readInt ()** - Returns a integer number from the stream.
- **.readLong ()** - Returns a long number from the stream. Long is a native Python data type, refer to <http://docs.python.org/> for more information.
- **.readStr (n)** - Returns a string from the stream based on the number of bytes to be read.
- **.seek (offset, whence)** - Moves the stream pointer position. Offset is the number of bytes to move, and whence specifies where to start moving from (0 = beginning of the stream, 1 = the current pointer location in the stream, 2 = the end of the stream).
- **.write (arg)** - Write to the stream. You can write to the stream using any of Python's native data types.
- **.Position** - Return the pointer position in the stream.
- **.Size** - Returns the byte size of the stream.

Refer to page 224 for general examples on how the stream classes are used.

Stream ()



Automation Systems Design only

Stream (stream)

The **Stream ()** base class allows *XSOL Automation* to directly manipulate stream types.



Do not use the **Stream ()** class to create a stream in an Expression.

The following properties and methods are available:

- **.copyFrom (stream)** - Copies the specified stream and returns the number of bytes copied.
- **.readFloat ()** - Returns a floating point number from the stream.
- **.readInt ()** - Returns a integer number from the stream.
- **.readLong ()** - Returns a long number from the stream. Long is a native Python data type, refer to <http://docs.python.org/> for more information.

- `.readStr (n)` - Returns a string from the stream based on the number of bytes to be read.
- `.seek (offset, whence)` - Moves the stream pointer position. Offset is the number of bytes to move, and whence specifies where to start moving from (0 = beginning of the stream, 1 = the current pointer location in the stream, 2 = the end of the stream).
- `.write (arg)` - Write to the stream. You can write to the stream using any of Python's native data types.
- `.Position` - Return the pointer position in the stream.
- `.Size` - Returns the byte size of the stream.

Refer to page 224 for general examples on how the stream classes are used.

StringStream ()



Automation Systems Design only

The `StringStream ()` class allows *XSOL Automation* to directly manipulate a string stream.

The following syntax forms are available:

- `StringStream (stream)` - Returns a `StringStream ()` instance from a stream.
- `StringStream (s)` - Returns a `StringStream ()` instance from a string.

The following properties and methods are available:

- `.GetDataString (stream)` - Returns the string value.
- `.SetDataString (stream)` - Set the string value.
- `.copyFrom (stream)` - Copies the specified stream and returns the number of bytes copied.
- `.readFloat ()` - Returns a floating point number from the stream.
- `.readInt ()` - Returns a integer number from the stream.
- `.readLong ()` - Returns a long number from the stream. Long is a native Python data type, refer to <http://docs.python.org/> for more information.
- `.readStr (n)` - Returns a string from the stream based on the number of bytes to be read.
- `.seek (offset, whence)` - Moves the stream pointer position. Offset is the number of bytes to move, and whence specifies where to start moving from (0 = beginning of the stream, 1 = the current pointer location in the stream, 2 = the end of the stream).
- `.write (arg)` - Write to the stream. You can write to the stream using any of Python's native data types.
- `.Position` - Return the pointer position in the stream.
- `.Size` - Returns the byte size of the stream.
- `.DataString` - Returns the string value.

Refer to page 224 for general examples on how the stream classes are used.

Examples

```

s = 'Acme Industries Incorporated'
ss1 = StringStream (s)           #Creates a string stream with the value of s
ss2 = ss1.GetDataString ()       #Returns Acme Industries Incorporated
ss3 = ss1.SetDataString ('Acme Holdings') #Sets the ss1 string to Acme Holdings
ss4 = ss1.DataString             #Returns Acme Holdings

```

8.1.19 User operations



Automation Systems Design only

The `UserRef ()` class returns a *XSOL Automation* user. It is used to refer to a user whenever a user is used in an Expression. It is mapped to XSOL's User type.



Do not use the `UserRef ()` class to create a user reference in an Expression.

The following properties are available:

- `.EmailAddress` - Returns the user's email address, as defined in Security Administration.
- `.FirstName` - Returns the user's first name, as defined in Security Administration.
- `.FullName` - Returns the user's full name, `FirstName` and `LastName`, as defined in Security Administration.
- `.LastName` - Returns the user's last name, as defined in Security Administration.
- `.UserID` - Returns the user's identification number.

Examples

This example assumes that `Sample_User` has an id of 2 and is John Drunner, johnd@acme.com.

```

u1 = Sample_User.EmailAddress #Returns johnd@acme.com
u2 = Sample_User.FirstName    #Returns John
u3 = Sample_User.FullName     #Returns John Drunner
u4 = Sample_User.LastName     #Returns Drunner
u5 = Sample_User.UserID       #Returns 2

```

8.1.20 Useful techniques



Automation Systems Design only

1. Checking that only one of a group of binary variables can be true

Assuming that `Var1`, `Var2`, `Var3` and `Var4` are binary variables, the expression would return True if one of `Var1` to `Var4` is True and the rest are False.

```
return sum (int(Var1), int(Var2), int(Var3), int(Var4)) == 1
```

2. A variant of (1) to check that AT MOST one variable can be True

```
return sum (int(Var1), int(Var2), int(Var3), int(Var4)) <= 1
```

3. False variants of (1) and (2)

Checking that one variable must be False:

```
return sum (int(not Var1), int(not Var2), int(not Var3), int(not Var4)) == 1
```

```
return sum (int(not Var1), int(not Var2), int(not Var3), int(not Var4)) <= 1
```

4. Checking several percentage variables so that none of them are less than a specified value

Assume Var1, Var2, Var3 and Var4 are percent variables and the expression would return True if any one of them is below 20%.

```
return min (Var1, Var2, Var3, Var4) < 20
```

5. As a variant of (4), check several percentage variables so that none of them are greater than 20%

```
return max (Var1, Var2, Var3, Var4) < 20
```

8.2 Command Line Parameters

The command line parameters listed below can be used to start *XSOL Mapping*. They are not case sensitive.

- `/n` - Starts *XSOL Mapping* but does not load, or ask to load, a model. This is used to always use an existing instance of *XSOL Mapping* when double-clicking on an `.esl` file, rather than opening a new instance.
- `path\model.esl` - Starts *XSOL Mapping* and loads the specified model.

Index

- .Area, 220
- .ConvertToUTCZone (), 205
- .Country, 220
- .Day, 204, 205
- .DayOfWeek, 204, 205
- .DayOfYear, 204, 205
- .Delimiter, 214
- .EmailAddress, 228
- .Ext, 214
- .Extension, 220
- .Filename, 214
- .Filepath, 214
- .FirstName, 228
- .FullName, 228
- .Hour, 205, 209
- .IsLeapYear (), 204, 205
- .LastName, 228
- .Minute, 205, 209
- .Month, 204, 205
- .Name, 212, 214
- .Number, 220
- .Path, 214
- .Second, 205, 209
- .Server, 212
- .StrPrecision (), 217, 219
- .User, 212
- .UserID, 228
- .Year, 204, 205
- .format (), 204, 205, 209
- .frac (), 216, 219
- .precision, 217, 219
- .round, 216, 219
- .toWords (), 216, 219
- .toWordsCents (), 216, 219
- .toWordsDollars (), 216, 219
- .toZeroPaddedString (), 216, 219
- Access level, 102
- AddFileToDocument (), 221
- adding
 - All, 83
 - Annotation, 88
 - Any, 83
 - attachments to Send Email, 67
 - Business Rule to a Reference Variable, 134
 - Companion Process, 50
 - Companion Process Variables to support external initiation, 52
 - Condition, 85
 - Control Group, 115
 - Datafield Variables, 135
 - datafields, 119
 - Decision, 85
 - Document Locations, 41
 - Document Locations limit, 42
 - documentation filter, 152
 - documentation layout, 154
 - Enterprise, 39
 - Expression, 138
 - Fork, 87
 - Group, 32, 88
 - HTML custom section, 157
 - image to a User Note, 146
 - Initiation Rule, 53
 - Joint, 83
 - limit to location, 42
 - link to a User Note, 146
 - Location, 68
 - Logical Printers, 40
 - Macro Expression, 138
 - model defined Control Group, 116
 - new Task icon, 173
 - Note, 145
 - Parameter, 141
 - Process, 48
 - Process Flow nodes, 80
 - Process Summaries, 74
 - Process Summary Filter, 77
 - Process Variables, 128
 - Reference Variable to a Business Rule, 134
 - Reference Variables, 133
 - Relationship, 118
 - Resource, 112
 - Resource Extension, 113
 - Role, 102
 - Secondary Index, 126
 - Stage, 54, 82
 - Start, 82
 - Stop, 82
 - Task, 60, 82

- Trading Entity, 39
- User Note, 145
- widget, 107
- Word 2003 custom section, 162
- Worksheet, 105
- aligning
 - Process Flow nodes, 93
 - widget, 110
 - widgets on a worksheet, 110
- alignment buttons, 110
- All, 83
 - adding node, 83
- Allow Negative, 46, 121, 122, 130, 131
- Allow Positive, 46, 121, 122, 130, 131
- Allow stage confirmation bypass, 55
- Allow stage to be manually parked?, 55
- Allow Zero, 46, 121, 122, 130, 131
- Alternate Text, 107
- Annotation, 88
 - adding node, 88
- Any, 83
 - adding node, 83
- appendDirectory (), 213
- appendix, 197
 - Python modifications, 197
- Archived after, 49
- Arithmetic (Process Variable), 129
- assigning
 - Expression, 140
 - Role, 102
- associations
 - Business Process constructs, 16
 - Control Groups, 21
 - Enterprise, 16
 - Groups, 17
 - Processes, 18
 - Relationships, 21
 - Resource Extensions, 20
 - Resources, 20
 - Roles, 17
 - Stages, 18
 - Tasks, 19
 - Trading Entities, 16
- attachments, 67
- Author, 34
- Auto Display Fields, 112
- Auto Increment (datafield), 120
- Auto Increment (Process Variable), 130
- Auto repeating process, 49
- average (), 200
- backing up, 37
 - model, 37
- bankersRounding, 216, 219
- Bevel (widget), 107
- Binary (), 202
- Binary (datafield), 121
- Binary (Process Variable), 130
- Binary (User Note), 45
- borders, 74
- Bring to front, 110
- bullets, 155
- Business Process constructs
 - associations, 16
 - definition, 16
 - examples, 16
- Business Process Management, 15
- Business Processes, 48
 - definition, 17
 - designing, 48
- Business Rule
 - adding to a Reference Variable, 134
 - limiting a Document Location, 42
 - selecting a user, 132
 - using with Reference Variables, 134
- Caption, 27
- certification test, 143
- certifying
 - all Expressions, 144
 - Expression, 143
 - Macro Expression, 143
- changing
 - Companion Process, 52
 - Control Group, 117
 - Datafield Variables, 135
 - datafields, 123
 - Document Locations, 42
 - documentation by selecting Roles, 153
 - Enterprise, 39
 - Escalation Level, 58
 - Expression, 140
 - Group, 33
 - Initiation Rule, 54
 - Logical Printers, 41
 - Macro Expression, 140
 - Parameter, 142
 - Process, 50
 - Process Flow default icon, 92
 - Process Flow icon, 91
 - Process Flow node properties, 95
 - Process Summaries, 76
 - Process Summary Filter, 79
 - Reference Variables, 134
 - Relationship, 119
 - Resource, 113
 - Resource Extension, 114
 - Role, 103

- Secondary Index, 127
- Stage, 55
- Stage display, 74
- Swim Lane orientation, 89
- Swim Lane's background colour, 89
- Swim Lane's width, 89
- Task, 73
- Task display, 74
- To Do List display, 60
- Trading Entity, 40
- User Note, 147
- Variables, 133
- widget, 109
- Worksheet, 106
- check entity, 149
- check model, 148
- checking
 - syntax of a Macro Expression, 142
 - syntax of an Expression, 142
- class
 - AddFileToDocument (), 221
 - Binary (), 202
 - Date (), 203
 - DateTime (), 204
 - Days (), 206
 - Document (), 211
 - DocumentLocation (), 212
 - EmailAddress (), 212
 - FileName (), 213
 - FileStream (), 224
 - FixedPoint (), 216
 - float (), 217
 - Hours (), 206
 - InstanceRef (), 221
 - int (), 217
 - MemoryStream (), 226
 - Minutes (), 207
 - Months (), 208
 - Percent (), 219
 - PhoneNumber (), 220
 - PrinterRef (), 220
 - Seconds (), 208
 - str (), 215
 - Stream (), 226
 - StringStream (), 227
 - Time (), 208
 - UserRef (), 228
 - WeekDay (), 209
 - WeekDayRange (), 210
 - Weeks (), 211
 - Years (), 211
- closing
 - Expansion Panel, 90
 - Task Process Flow, 90
- coalesce (), 200
- Code (datafield), 121
- columns, 29
 - sorting, 29
- command line, 230
 - parameters, 230
- common Worksheet, 105
- Companion Process, 50
 - adding, 50
 - adding Variables to support external initiation, 52
 - changing, 52
 - sharing Process Variables, 52
- Condition, 85
 - adding node, 85
- constant
 - CURRENT_ZONE, 203, 205
 - UTC_ZONE, 203, 205
- constructs
 - associations, 16
 - definition, 16
 - examples, 16
- context-sensitive menus, 28
- Control Group, 115
 - adding, 115
 - associations, 21
 - changing, 117
 - definition, 21
 - examples, 21
 - using, 115
- Control Group (datafield), 121
- Copy from source, 31
- copying, 30
 - to a User Note from Office, 146
- count (), 201
- countBlank (), 201
- Country Code Mandatory, 122, 131
- Create Document from a Template, 61
- Created at, 45
- creating
 - model, 34
 - Process Flow, 80
 - Tables, 112
- CSS, 160
 - using in HTML output, 160
- Current user, 43
- CURRENT_ZONE, 203, 205
- custom section
 - adding HTML, 157
 - adding Word 2003, 162
 - editing HTML, 157
 - editing Word 2003, 162
 - inserting HTML, 157
 - inserting Word 2003, 162

- removing HTML, 157
- removing Word 2003, 162
- renaming HTML, 157
- renaming Word 2003, 162
- customising
 - designers, 27
- Datafield Variable
 - adding, 135
 - changing, 135
 - using, 135
- datafields, 112, 119, 121
 - adding, 119
 - Allow Negative, 121, 122
 - Allow Positive, 121, 122
 - Allow Zero, 121, 122
 - Auto Increment, 120
 - Binary, 121
 - Calculated, 120
 - Calculation, 120
 - Caption, 120
 - changing, 123
 - Code, 121
 - Control Group, 121
 - Country Code Mandatory, 122
 - Data Type, 120
 - Datafield, 121
 - Datafield default, 120
 - Date, 121
 - Datetime, 121
 - Days of the Week, 130
 - Default Logical Printer, 122
 - Display Mask, 120
 - Document, 121
 - Email Address, 121
 - End At, 120
 - External ID, 120
 - False Value, 121
 - Index, 121
 - Input Mask, 120
 - Length, 121
 - Mandatory, 120
 - Memo, 121
 - No of Decimal Digits, 121, 122
 - No of Integer Digits, 121
 - Notational, 121
 - Numeric, 121
 - Parent Lookup, 122
 - Percent, 122
 - Phone Number, 122
 - Printer, 122
 - properties, 119
 - Repeat Count, 120
 - Run-time Hint, 120
 - Scientific, 122
 - Start At, 120
 - Step, 120
 - Time, 122
 - True Value, 121
 - User, 122
 - User Notes, 120
 - using, 119
- Date (), 203
- Date (datafield), 121
- Date (Process Variable), 130
- Date (User Note), 45
- Datetime (), 204
- Datetime (datafield), 121
- Datetime (Process Variable), 130
- Datetime (User Note), 45
- Days (), 206
- Days of the Week (datafield), 130
- DaysDiff (), 206
- Decision, 85
 - adding node, 85
- Default 'false' value, 35
- Default 'null' value, 35
- Default 'show role assignments' value, 35
- Default 'true' value, 35
- Default access level, 56
- Default caption for new 'False' links, 35
- Default caption for new 'True' links, 35
- Default Execution Level, 48
- Default execution level, 53
- Default flow control icons, 35
- Default height for annotated flow nodes, 34
- Default height for other flow nodes, 34
- Default Logical Printer, 122, 131
- Default rotation for flow diagrams, 34
- Default stage flow 'Print in documentation', 35
- Default task icons, 35
- Default Unit of Measure, 34
- Default width for annotated flow nodes, 34
- Default width for other flow nodes, 34
- defining
 - Note, 45
 - User Note, 45
 - Variables that map to an initiated Process, 50
- definition
 - Business Process constructs, 16
 - Business Processes, 17
 - Control Groups, 21
 - Enterprise, 16
 - Process Flow, 19
 - Processes, 18
 - Relationships, 21
 - Resource Extensions, 20
 - Resources, 20

- Roles, 17
- Stages, 18
- Tables, 20
- Tasks, 18
- Trading Entities, 16
- Deleted at, 45
- deleteFile (), 213
- deleting
 - any item, 33
 - documentation filter, 152
 - documentation layout, 154
 - Enterprise, 39
 - model defined Control Group, 117
 - Role, 103
 - Task, 73
- deleting items, 33
- Design tasks - process variables map, 62
- designer, 27
 - customising, 27
 - displaying, 27
 - hiding, 27
 - ordering, 27
 - using, 26
- Designer comments, 28
- designing
 - Business Processes, 48
 - Enterprise, 39
 - model, 34
 - Process Flow, 80
 - Role, 102
 - Tables, 112
 - To Do List display, 58
 - To Do List sort order, 59
 - Trading Entity, 39
 - Variables, 128
- Diagram Magnification, 167
- display mask
 - syntax, 125
 - using, 125
- displaying
 - designers, 27
 - Task icons, 173
- Document (), 211
- Document (datafield), 121
- Document (Process Variable), 130
- Document Location, 41
 - adding, 41
 - adding a limit, 42
 - changing, 42
- Document Operation, 61
- documentation, 151
 - generating, 151
 - generating outputs, 151
 - HTML, 151
 - HTML Help compiled help file, 151
 - Microsoft Word, 151
 - setting up HTML Help, 161
 - spreadsheet, 151
 - web site, 151
- documentation filter, 152
 - adding, 152
 - deleting, 152
 - editing, 152
 - renaming, 152
 - using, 152
- documentation layout, 154
 - adding, 154
 - deleting, 154
 - editing, 154
 - renaming, 154
 - using, 154
- documentation template
 - editing HTML, 159
 - editing Word 2003, 164
 - using HTML, 159
 - using Word 2003, 164
- DocumentLocation (), 212
- Draw 'True' links first?, 35
- drill-downs, 74
- editing
 - documentation filter, 152
 - documentation layout, 154
 - HTML custom section, 157
 - HTML documentation template, 159
 - model defined Control Group, 116
 - Word 2003 custom section, 162
 - Word 2003 documentation template, 164
- Email Address (datafield), 121
- Email Address (Process Variable), 130
- EmailAddress (), 212
- emailing
 - HTML documentation, 161
 - model, 36
- End At, 120, 130
- End-User License Agreement, 191
- Enterprise, 39
 - adding, 39
 - associations, 16
 - changing, 39
 - definition, 16
 - deleting, 39
 - designing, 39
 - examples, 16
- Enterprise name, 43
- Escalation Level, 56
 - changing, 58
 - handling exceptions, 57

- using, 56
- EULA, 191
- examples
 - Business Process constructs, 16
 - Control Groups, 21
 - Enterprise, 16
 - Groups, 17
 - Process Flow, 19
 - Processes, 18
 - Relationships, 21
 - Resource Extensions, 20
 - Resources, 20
 - Roles, 17
 - Stages, 18
 - Tables, 20
 - Tasks, 19
 - Trading Entities, 16
- Excel, 151
- exception flow, 57
 - using, 57
- exists (), 213
- Expansion Panel, 90, 96
 - closing, 90
 - opening, 90
- expiration date, 190
- Expression, 138
 - adding, 138
 - assigned to Variable, 128
 - assigning, 140
 - certifying, 143
 - certifying all, 144
 - changing, 140
 - checking syntax, 142
 - Condition and Decision nodes, 138
 - Macro Expressions, 138
 - not assigned to Variable, 128
 - Schedule Expression, 138
 - Stage Summary, 138
 - Time Calculation, 138
 - Variables, 138
- False Value, 45, 121, 130
- File menu, 28
- FileName (), 213
- FileStream (), 224
- filtering
 - Process Flow by Role, 98
 - Role, 103
- First Day of the Week, 40
- FixedPoint (), 216
- float (), 217
- flyovers, 148
- forceDirectories (), 214
- Fork, 87
 - adding node, 87
 - lengthen, 88
 - rotating, 88
 - shorten, 88
- function
 - appendDirectory (), 213
 - average (), 200
 - coalesce (), 200
 - count (), 201
 - countBlank (), 201
 - DaysDiff (), 206
 - deleteFile (), 213
 - exists (), 213
 - forceDirectories (), 214
 - getTaskDataXml (), 221
 - hasNoValue (), 201
 - hasValue (), 201
 - HoursDiff (), 207
 - max (), 201
 - min (), 202
 - MinutesDiff (), 207
 - move (), 214
 - pathList (), 215
 - removeDirectories (), 215
 - SecondsDiff (), 208
 - sum (), 202
 - toString (), 202
 - WeeksDiff (), 211
- generating
 - documentation, 151
 - documentation outputs, 151
 - links automatically, 90
 - Start/Stop nodes automatically, 90
- getTaskDataXml (), 221
- grid lines, 96
- Group, 32, 88
 - adding, 32
 - adding node, 88
 - associations, 17
 - changing, 33
 - definition, 17
 - examples, 17
- Grouping (widget), 107
- handling
 - Escalation Level exceptions, 57
- hasNoValue (), 201
- hasValue (), 201
- health check, 24
- help, 13, 14
 - using, 14
- Help menu, 28
- Help URL, 28

- hiding
 - designers, 27
 - Task icons, 173
- Horizontal Rule (widget), 107
- Hours (), 206
- HoursDiff (), 207
- HTML, 151
 - CSS, 160
 - styles, 160
 - using styles (CSS), 160
- HTML documentation
 - emailing, 161
- HTML Help
 - setting up, 161
- HTML Help compiled help file, 151
- Image (widget), 107
- Image Repeat, 107
- Important, 14
- Index (datafield), 121
- Initiate process, 62
- Initiation Rule
 - adding, 53
 - changing, 54
- input mask
 - syntax, 123
- Insert menu, 28
- inserting
 - HTML custom section, 157
 - Word 2003 custom section, 162
- InstanceRef (), 221
- int (), 217
- introduction, 13
- Joint, 83
 - adding node, 83
- key, 14
- keyboard shortcuts, 180
- keywords, 198
- Label (widget), 107
- large diagrams, 168
- Last updated at, 45
- layout, 167
- Length (datafield), 121
- Length (Process Variable), 131
- Length (User Note), 46
- limit to location, 42
 - adding, 42
- limiting
 - Document Location with a Business Rule, 42
- linking
 - nodes automatically, 90
 - Process Flow nodes, 94
 - linking to HTML documentation, 162
- loading
 - model, 37
- Location
 - adding, 68
- logging off, 22
- logging on, 22
- Logical Printer, 40
 - adding, 40
 - changing, 41
- Macro Expression, 138
 - adding, 138
 - certifying, 143
 - changing, 140
 - checking syntax, 142
- Maintenance Task Worksheet, 105
- Manual, 62
- manual
 - using, 14
- mapping
 - Variables from one Process to another, 66
- max (), 201
- Maximum flow node height, 34
- Maximum flow node width, 34
- Memo (datafield), 121
- Memo (Process Variable), 131
- Memo (User Note), 45
- MemoryStream (), 226
- menus, 28
 - context-sensitive, 28
 - File, 28
 - Help, 28
 - Insert, 28
 - Process, 28
 - Tools, 28
 - View, 28
 - Window, 28
- method
 - .ConvertToUTCZone (), 205
 - .IsLeapYear (), 204, 205
 - .StrPrecision (), 217, 219
 - .format (), 204, 205, 209
 - .frac (), 216, 219
 - .toWords (), 216, 219
 - .toWordsCents (), 216, 219
 - .toWordsDollars (), 216, 219
 - .toZeroPaddedString (), 216, 219
- Microsoft Word, 151
 - styles, 165
 - using style templates, 165
 - using styles, 165
- min (), 202
- Minutes (), 207

- MinutesDiff (), 207
- model, 34
 - removing version information, 38
 - backing up, 37
 - creating, 34
 - designing, 34
 - emailing, 36
 - loading, 37
 - opening, 37
 - packaging, 36
 - properties, 34
 - restoring, 37
 - saving, 35
 - unpackaging, 37
 - validating, 148
- Model committed at, 43
- model defined Control Group
 - adding, 116
 - deleting, 117
 - editing, 116
- Model Defined Entries, 115
- Model version, 43
- Months (), 208
- move (), 214
- moving
 - Swim Lane, 89
- multiple-user environment, 23
- navigation, 26
- Negate the override?, 56
- No of Decimal Digits, 46, 121, 122, 130, 131
- No of Integer Digits, 46, 121, 129, 131
- node alignment buttons, 110
- nodes, 80
 - All, 83
 - Annotation, 88
 - Any, 83
 - Condition, 85
 - Decision, 85
 - Fork, 87
 - Group, 88
 - Joint, 83
 - Stage, 82
 - Start, 82
 - Stop, 82
 - Task, 82
- Non-XSOL, 62
- Normal style, 147
- Normal.dot, 147
- normalRounding, 216, 219
- Notational (datafield), 121
- Notational (Process Variable), 131
- Notational (User Note), 46
- Note, 14, 145
 - adding, 145
 - defining, 45
 - using, 145
- Null or unknown value, 45, 121, 130
- numbering, 155
- numbers, 155
- Numeric (datafield), 121
- Numeric (Process Variable), 131
- Numeric (User Note), 46
- Office template, 69
 - using in a Task, 69
- opening
 - Expansion Panel, 90
 - model, 37
 - Task Process Flow, 90
- operations
 - binary, 202
 - email, 212
 - file, 213
 - notational, 215
 - number, 215
 - percent, 218
 - phone, 220
 - printer, 211, 220
 - Reference Variable, 221
 - Script, 221
 - Stream, 224
 - user, 228
- options, 169
 - Default Icons, 178
 - Documentation, 175
 - Editor Colours, 177
 - Editor Options, 176
 - General, 169
 - Highlighting, 171
 - Icons, 173
 - New Model Defaults, 170
 - Predefined Filters, 176
- ordering
 - designers, 27
- Override with, 53, 56
- packaging
 - model, 36
- page boundaries, 96, 167
- page layout, 167
- Panel Height, 105
- Panel Width, 105
- Paper Size, 34, 167, 168
- Parameter, 141
 - adding, 141
 - changing, 142
- Parent Lookup (datafield), 122

- Paste, 30
- Paste Link, 30
- pasting, 30
- pathList (), 215
- Percent (), 219
- Percent (datafield), 122
- Percent (Process Variable), 131
- Percent (User Note), 46
- Performed by System, 55
- Permanently deleted after, 49
- Phone Number (datafield), 122
- Phone Number (Process Variable), 131
- PhoneNumber (), 220
- Print Document, 62
- Printer (datafield), 122
- Printer (Process Variable), 131
- PrinterRef (), 220
- printing
 - Diagram Magnification, 167
 - large diagrams, 168
 - Paper Size, 167
 - Process Flow diagram, 99
 - Rotation for flow diagrams, 167
 - tips, 167
- printing options for documentation generation
 - Process Flow, 100
- problem, 13
- Process, 48
 - adding, 48
 - associations, 18
 - changing, 50
 - defining Variables that map to an initiated, 50
 - definition, 18
 - examples, 18
 - using, 48
- Process archived?, 43
- Process finished at, 43
- Process Flow, 80
 - adding nodes, 80
 - aligning nodes, 93
 - changing default icon, 92
 - changing icon, 91
 - changing node properties, 95
 - creating, 80
 - definition, 19
 - designing, 80
 - examples, 19
 - filtering by Role, 98
 - linking nodes, 94
 - printing a diagram, 99
 - printing options for documentation generation, 100
 - saving a diagram, 99
 - selecting nodes, 96
 - validating links, 95
 - viewing nodes, 96
 - viewing Role information, 96
- Process Flow nodes, 80
- Process initiated by Role, 43
- Process initiated by Team, 43
- Process initiated by user, 43
- Process instance id, 43
- Process last updated at, 43, 44
- Process last updated by user, 43, 44
- Process manually initiated?, 43
- Process menu, 28
- Process name, 44
- Process started at, 44
- Process state, 44
- Process Summary, 74
 - adding, 74
 - changing, 76
 - drill-downs, 74
 - using, 74
- Process Summary Filter, 77
 - adding, 77
 - changing, 79
 - using, 77
- Process Summary Worksheet, 105
- Process Update, 63
- Process Variables, 128
- Process version, 44
- properties
 - Caption, 27
 - changing Process Flow, 95
 - datafields, 119
 - Designer comments, 28
 - Help URL, 28
 - model, 34
 - Runtime Hint, 27
 - User Notes, 27
 - Version information, 28
- Properties Editor, 27
- property
 - .Area, 220
 - .Country, 220
 - .Day, 204, 205
 - .DayOfWeek, 204, 205
 - .DayOfYear, 204, 205
 - .Delimiter, 214
 - .EmailAddress, 228
 - .Ext, 214
 - .Extension, 220
 - .Filename, 214
 - .Filepath, 214
 - .FirstName, 228
 - .FullName, 228

- .Hour, 205, 209
- .LastName, 228
- .Minute, 205, 209
- .Month, 204, 205
- .Name, 212, 214
- .Number, 220
- .Path, 214
- .Second, 205, 209
- .Server, 212
- .User, 212
- .UserID, 228
- .Year, 204, 205
- .precision, 217, 219
- .round, 216, 219
- bankersRounding, 216, 219
- normalRounding, 216, 219
- Python, 197
 - basic types, 197
 - binary operations, 202
 - date/time operations, 203
 - email operations, 212
 - global functions, 200
 - naming conventions, 197
 - non-standard syntax, 198
 - number operations, 213, 215
 - percent operations, 218
 - phone, 220
 - printer operations, 211, 220
 - Reference Variable operations, 221
 - reserved keywords, 198
 - Script operations, 221
 - Stream operations, 224
 - user operations, 228
- quick keys, 180
- Ranges, 46, 122, 130, 131
- ranges, 118
- Record created by user, 45
- Record deleted by user, 45
- Record instance id, 45
- Record instance version, 45
- Record last updated by user, 45
- redoing, 30
- Reference, 48, 54
- Reference Caption Position, 34
- Reference Caption Separator, 34
- reference captions, 100
 - setting up, 100
 - using, 100
- Reference Variable
 - adding, 133
 - adding to a Business Rule, 134
 - changing, 134
 - using, 133
 - using with a Business Rule, 134
- registration, 190
- Relationship, 118
 - adding, 118
 - associations, 21
 - changing, 119
 - definition, 21
 - examples, 21
 - using, 118
- Removed after, 49
- removeDirectories (), 215
- removing
 - HTML custom section, 157
 - Word 2003 custom section, 162
- removing
 - model version information, 38
- renaming
 - documentation filter, 152
 - documentation layout, 154
 - HTML custom section, 157
 - Word 2003 custom section, 162
- reserved keywords, 198
- Reset individual node icons, 35
- resizing
 - items, 29
- Resource, 112
 - adding, 112
 - associations, 20
 - changing, 113
 - definition, 20
 - examples, 20
 - using, 112
- Resource Extension, 113
 - adding, 113
 - associations, 20
 - changing, 114
 - definition, 20
 - examples, 20
 - using, 113
- restoring
 - model, 37
- Result Type, 138
- Role, 102
 - adding, 102
 - assigning, 102
 - associations, 17
 - changing, 103
 - definition, 17
 - deleting, 103
 - designing, 102
 - examples, 17
 - filtering, 103
 - filtering Process Flow by, 98

- rotate Forks, 88
- Rotation for flow diagrams, 167
- Runtime Hint, 27
- Save or Reset application defaults, 34
- saving
 - model, 35
 - Process Flow diagram, 99
- Schedule Expression, 54, 138
- Scientific (datafield), 122
- Scientific (Process Variable), 131
- Scientific (User Note), 46
- Script, 63
- Secondary Index
 - adding, 126
 - changing, 127
 - using, 126
- Seconds (), 208
- SecondsDiff (), 208
- Select, 80
- selecting
 - document in XSOL Automation, 67
 - Process Flow nodes, 96
 - Roles changes generated documentation, 153
 - user in XSOL Automation, 132
 - user with a Business Rule, 132
- Send Email, 63
 - adding attachments, 67
- Send Fax, 64
- Send Pager Message, 64
- Send SMS Message, 64
- Send to back, 110
- setting
 - common Worksheet, 105
 - Default Icons options, 178
 - Documentation options, 175
 - Editor Colours options, 177
 - Editor Options options, 176
 - General options, 169
 - Highlighting options, 171
 - Icons options, 173
 - New Model Defaults options, 170
 - Predefined Filters options, 176
 - Tab order, 111
- setting up
 - HTML Help, 161
 - reference captions, 100
- sharing
 - Companion Process Variables to Processes, 52
- shortcuts, 180
- Show grid, 110
- Show stage border, 74
- Show stage icon, 74
- Show task border, 74
- Show task icon, 74
- Snap to grid, 110
- sorting
 - columns, 29
- spreadsheet, 151
- Stage, 54, 82
 - adding, 54
 - adding node, 82
 - associations, 18
 - borders, 74
 - changing, 55
 - changing display, 74
 - closing Process Flow, 90
 - definition, 18
 - examples, 18
 - icons, 74
 - opening Process Flow, 90
 - using, 54
- Stage completion due, 44
- Stage escalation level, 44
- Stage finished at, 44
- Stage name, 44
- Stage started at, 44
- Stage state, 44
- Stage Summary, 138
- Stage Summary for To-do List, 48, 54
- Stage undertaken by role, 44
- Stage undertaken by team, 44
- Stage undertaken by user, 44
- Stage version, 44
- Start, 82
 - adding node, 82
 - creating automatically, 90
- Start At, 120, 130
- Start month of financial year, 39
- Step, 120, 130
- Stop, 82
 - adding node, 82
 - creating automatically, 90
- Stored Variable, 128
- str (), 215
- Stream (), 226
- StringStream (), 227
- sum (), 202
- support, 13
- swap links, 85
- Swim Lane, 89
 - changing background colour, 89
 - changing orientation, 89
 - changing width, 89
 - moving, 89
 - using, 89
- syntax
 - display mask, 125

- input mask, 123
- System, 65
- System Calculation, 65
- System Values, 43
 - Created at, 45
 - Current user, 43
 - Deleted at, 45
 - Enterprise name, 43
 - Last updated at, 45
 - Model committed at, 43
 - Model version, 43
 - Process archived?, 43
 - Process finished at, 43
 - Process initiated by Role, 43
 - Process initiated by Team, 43
 - Process initiated by user, 43
 - Process instance id, 43
 - Process last updated at, 43, 44
 - Process last updated by user, 43, 44
 - Process manually initiated?, 43
 - Process name, 44
 - Process started at, 44
 - Process state, 44
 - Process version, 44
 - Record created by user, 45
 - Record deleted by user, 45
 - Record instance id, 45
 - Record instance version, 45
 - Record last updated by user, 45
 - Stage completion due, 44
 - Stage escalation level, 44
 - Stage finished at, 44
 - Stage name, 44
 - Stage started at, 44
 - Stage state, 44
 - Stage undertaken by role, 44
 - Stage undertaken by team, 44
 - Stage undertaken by user, 44
 - Stage version, 44
 - Table definition version, 45
 - Table name, 45
 - Task finished at, 44
 - Task name, 44
 - Task started at, 44
 - Task state, 44
 - Task type, 44
 - Task undertaken by role, 44
 - Task undertaken by team, 45
 - Task undertaken by user, 45
 - Task version, 45
 - Trading entity name, 43
 - Trading entity version, 43
 - using, 43

- Tab order, 111
 - setting, 111
- Table, 112
 - creating, 112
 - definition, 20
 - designing, 112
 - examples, 20
- Table definition version, 45
- Table name, 45
- Table Update, 65
- Task, 60, 82
 - adding, 60
 - adding attachments, 67
 - adding new icon, 173
 - adding node, 82
 - associations, 19
 - borders, 74
 - changing, 73
 - changing display, 74
 - Create Document from a Template, 61
 - definition, 18
 - deleting, 73
 - Document Operation, 61
 - examples, 19
 - icons, 74
 - Initiate process, 62
 - Manual, 62
 - Non-XSOL, 62
 - Print Document, 62
 - Process Update, 63
 - Script, 63
 - Send Email, 63
 - Send Fax, 64
 - Send Pager Message, 64
 - Send SMS Message, 64
 - System, 65
 - System Calculation, 65
 - Table Update, 65
 - using, 60
 - using Office templates, 69
 - Worksheet, 65
- Task finished at, 44
- Task icon associations, 174
- Task name, 44
- Task started at, 44
- Task state, 44
- Task type, 44
- Task types, 60
- Task undertaken by role, 44
- Task undertaken by team, 45
- Task undertaken by user, 45
- Task version, 45
- Task Worksheet, 105
- templates

- using Word 2000/XP, 165
- Time (), 208
- Time (datafield), 122
- Time (Process Variable), 131
- Time (User Note), 46
- Time Calculation, 138
- Time Zone, 40
- Timeout, 56
- tips
 - printing, 167
- title page, 168
- To Do List, 58
 - changing display, 60
 - designing display, 58
 - designing sort order, 59
- To-do Values Action, 55
- Tools menu, 28
- toString (), 202
- Trading Entity, 39
 - adding, 39
 - associations, 16
 - changing, 40
 - definition, 16
 - designing, 39
 - examples, 16
- Trading entity name, 43
- Trading entity version, 43
- tree view
 - close, 189
 - collapse, 189
 - expand, 189
 - open, 189
- True Value, 45, 121, 130
- undoing, 30
- unpackaging
 - model, 37
- Url, 107
- User (datafield), 122
- User (Process Variable), 131
- User Note, 145
 - adding, 145
 - adding a link, 146
 - adding an image, 146
 - Allow Negative, 46
 - Allow Positive, 46
 - Allow Zero, 46
 - Binary, 45
 - changing, 147
 - copying information from Office, 146
 - Date, 45
 - Datetime, 45
 - defining, 45
 - False Value, 45
 - Length, 46
 - Memo, 45
 - No of Decimal Digits, 46
 - No of Integer Digits, 46
 - Notational, 46
 - Null or unknown value, 45
 - Numeric, 46
 - Percent, 46
 - Scientific, 46
 - Time, 46
 - True Value, 45
 - using, 145
 - Word, 45
- User Notes, 27
- UserRef (), 228
- using
 - Business Rule with Reference Variables, 134
 - Control Group, 115
 - Datafield Variables, 135
 - datafields, 119
 - designers, 26
 - display mask, 125
 - documentation filter, 152
 - documentation layout, 154
 - Escalation Level, 56
 - exception flow, 57
 - help, 14
 - HTML documentation template, 159
 - HTML styles, 160
 - manual, 14
 - Note, 145
 - Office templates in a Task, 69
 - Process, 48
 - Process Summaries, 74
 - Process Summary Filter, 77
 - Process Variables, 128
 - reference captions, 100
 - Reference Variable with a Business Rule, 134
 - Reference Variables, 133
 - Relationship, 118
 - Resource, 112
 - Resource Extension, 113
 - Secondary Index, 126
 - Stage, 54
 - Swim Lanes, 89
 - System Values, 43
 - Task, 60
 - Task icon associations, 174
 - User Note, 145
 - widget, 107
 - Word 2003 documentation template, 164
 - Word User Notes, 146
 - Worksheet, 105
 - XSOL Mapping, 22

- UTC_ZONE, 203, 205
- validating
 - model, 148
 - Process Flow links, 95
 - specific items, 149
- validation, 148
- Variable, 128
 - adding Process, 128
 - Allow Negative, 130, 131
 - Allow Positive, 130, 131
 - Allow Zero, 130, 131
 - Arithmetic, 129
 - Auto Increment, 130
 - Binary, 130
 - Caption, 129
 - changing, 133
 - Country Code Mandatory, 131
 - Data Type, 129
 - Date, 130
 - Datetime, 130
 - Default Logical Printer, 131
 - defining that map to an initiated Process, 50
 - designing, 128
 - Display Mask, 129
 - Document, 130
 - Email Address, 130
 - End At, 130
 - Expressions, 138
 - External ID, 129
 - False Value, 130
 - Input Mask, 129
 - Length, 131
 - mapping from one Process to another, 66
 - Memo, 131
 - No of Decimal Digits, 130, 131
 - No of Integer Digits, 129, 131
 - Notational, 131
 - Null or unknown value, 121, 130
 - Numeric, 131
 - Percent, 131
 - Phone Number, 131
 - Printer, 131
 - Repeat Count, 129
 - Run-time Hint, 129
 - Scientific, 131
 - Start At, 130
 - Step, 130
 - Store variable, 129
 - Stored, 128
 - Time, 131
 - True Value, 130
 - User, 131
 - User Notes, 129
 - using Process, 128
 - Variable default, 129
 - Work, 128
- Variables to Update to Parent Process, 63
- Version information, 28
- version information, 38
- View menu, 28
- viewing
 - available Processes, 97
 - available Reference Variables, 97
 - Process Flow nodes, 96
- Warning, 14
- web sites, 151
- WeekDay (), 209
- WeekDayRange (), 210
- Weeks (), 211
- WeeksDiff (), 211
- Where Used, 150
- widget, 107
 - adding, 107
 - aligning, 110
 - Bevel, 107
 - changing, 109
 - Grouping, 107
 - Horizontal Rule, 107
 - Image, 107
 - Label, 107
 - properties, 108
 - using, 107
- Window menu, 28
- Word (User Note), 45
- Work Variable, 128
- Worksheet, 65, 105
 - adding, 105
 - changing, 106
 - common, 105
 - using, 105
- worksheet
 - aligning widgets, 110
 - setting Tab order, 111
- XSOL Automation
 - selecting a document, 67
 - selecting a User, 132
- XSOL Mapping, 13
 - multiple-user environment, 23
 - navigation, 26
 - using, 22
- Years (), 211
- zooming, 96