XSOL Automation

Method & Delivery

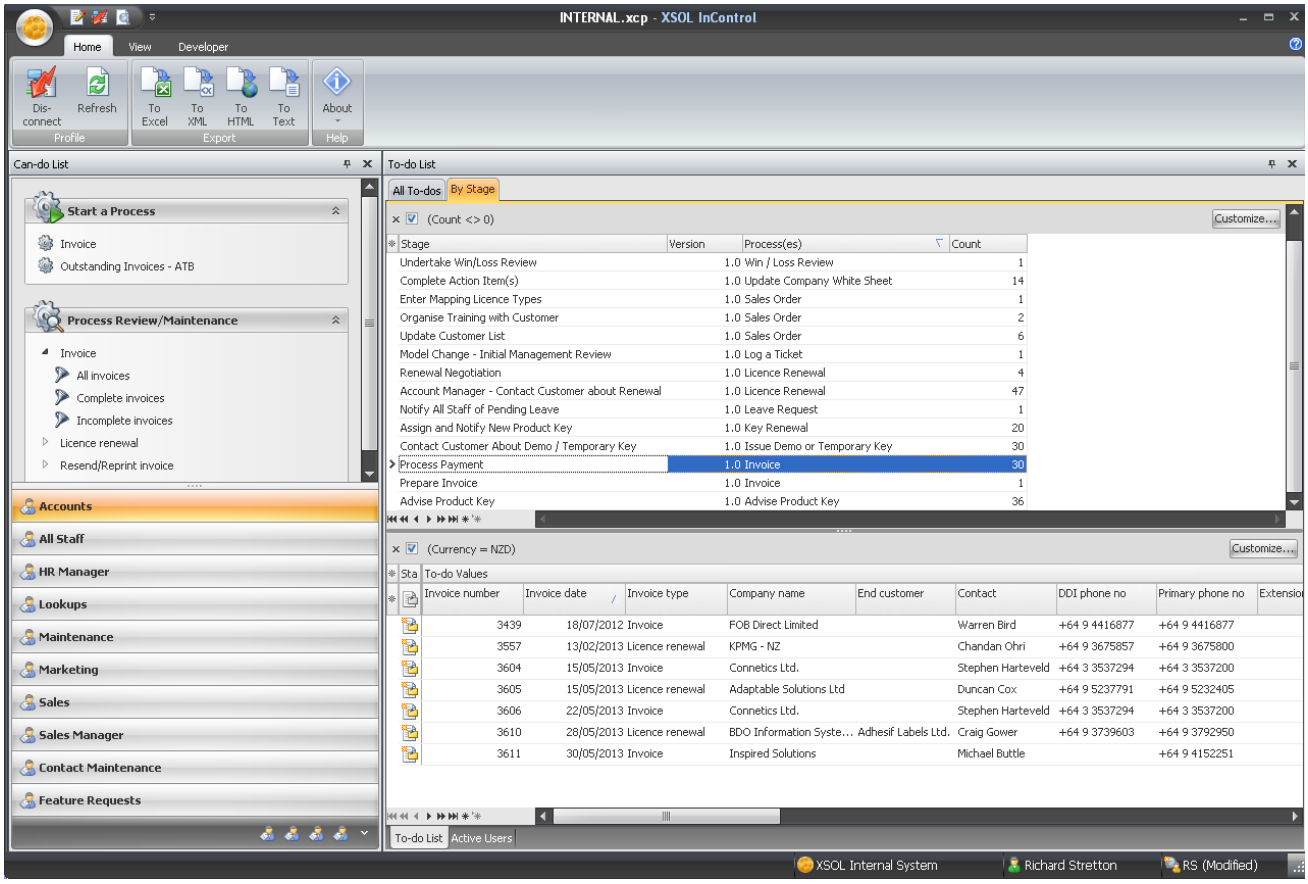
THE USER EXPERIENCE

Users interact with an XSOL Automation system through a Client control panel. On the left side of the panel is a list of the Roles that the user has been allocated and the processes that they can initiate (**Can-Do List**) as specified within the Design Process Model (see later). Roles may enable users to:

- Begin an instance of a Process
- View Process Summary Dashboard Queries showing a range of information about running or complete process instances (as defined in the model).
- View SQL reference data via Lookup Lists
- Modify SQL reference data via Maintenance tasks

The central area of the screen (**To-Do List**) shows the parts of active processes (Stages) that the user is required to undertake as part of their Role. Work can be displayed as individual Stages or as a list of all To-Do items. The dashboard is configurable by the end user, to filter out items they do not require.

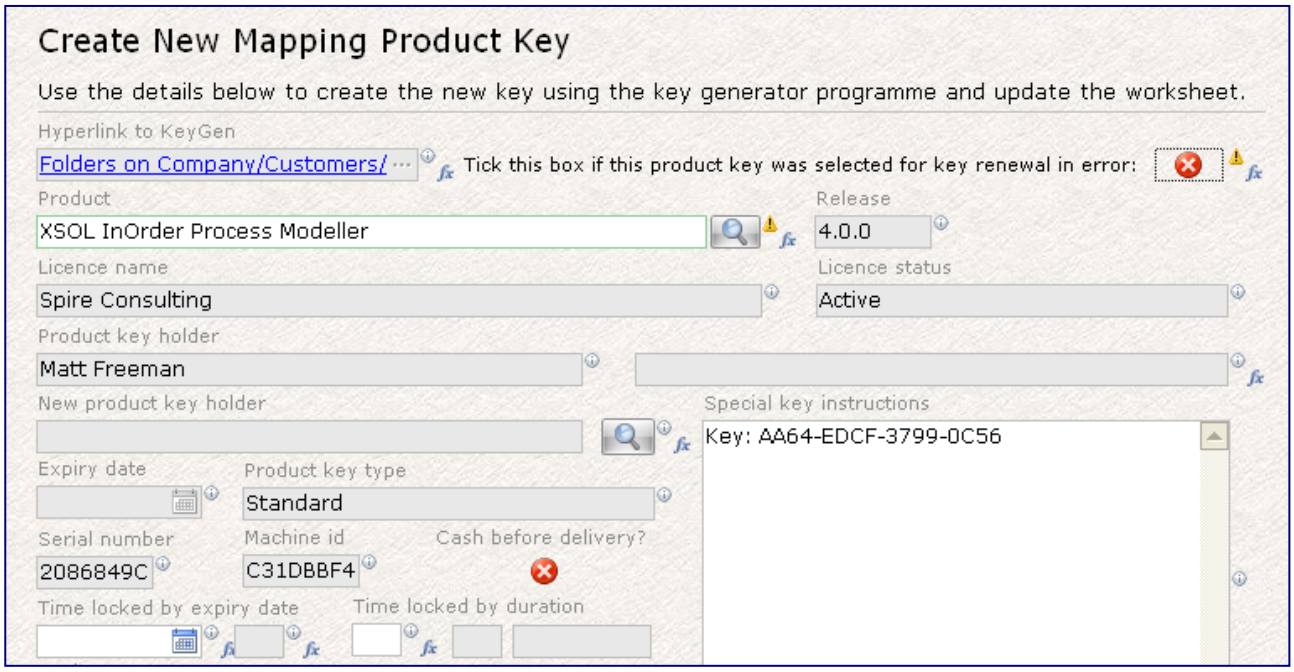
Below: A typical user's desktop in a complex XSOL Automation system.

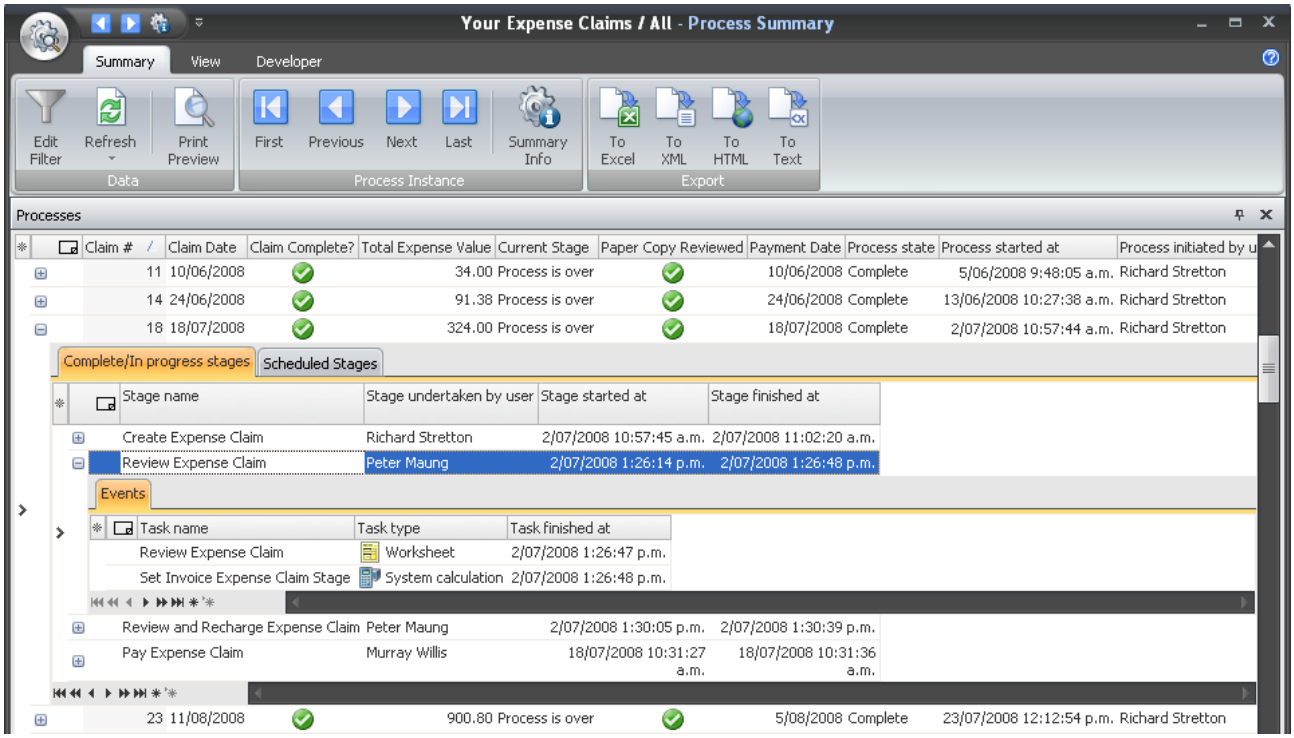


Users primarily interact with running processes by reading and entering information into Worksheet Tasks (as designed in the original model file).

Process Summaries show processes that have completed or are running. The user can drill down through the information to see exactly what has happened with any process instance and who was responsible for it. Summaries can be customized and the resulting 'Views' saved to create dashboards of process information.

Below: An example of a Worksheet at runtime.



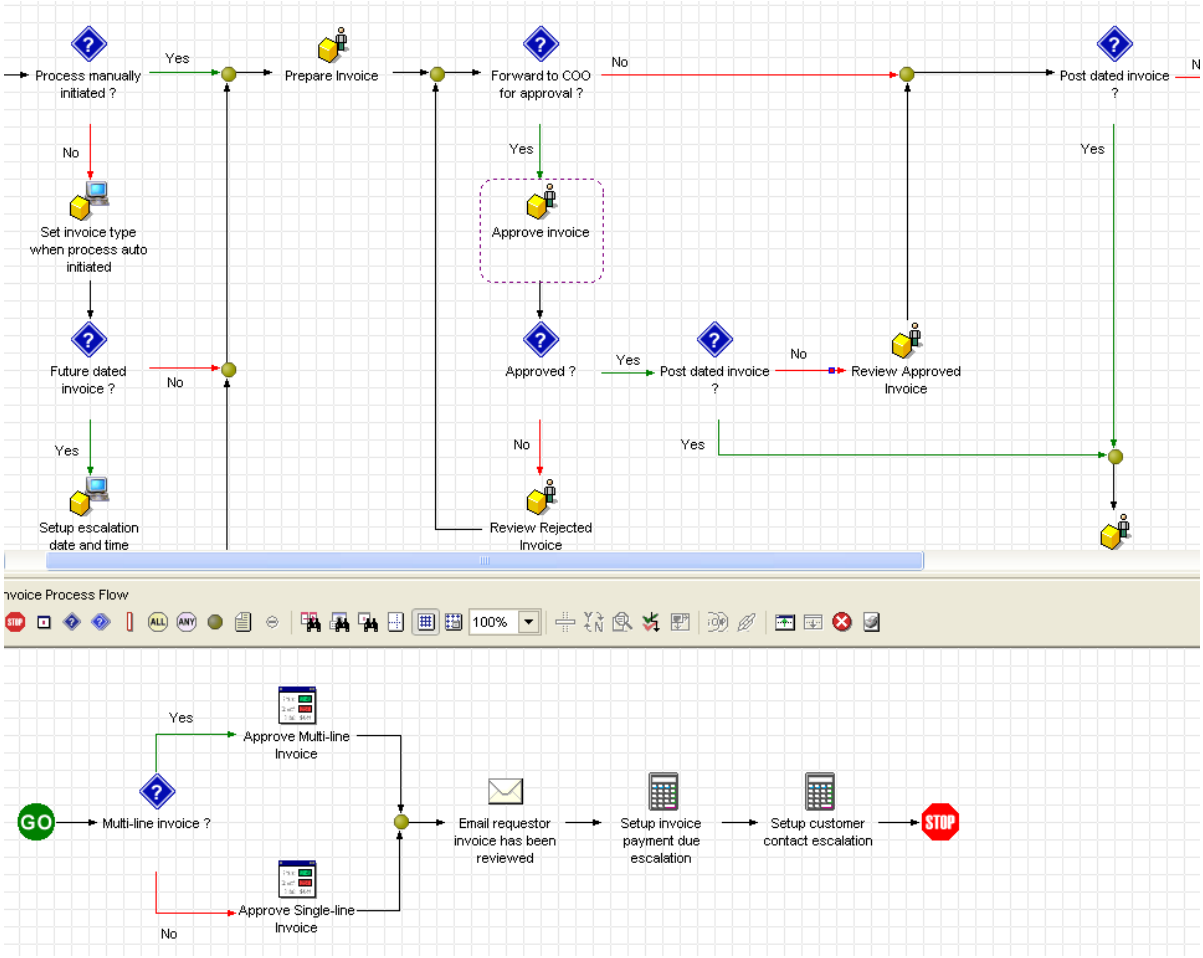


Above: An example of a Process Summary where a single process instance has been opened and the user has drilled down to the Task level. Information concerning when the Stage was started, finished and by whom is visible, as are a variety of variables (as specified in the model).

PROCESS AUTOMATION DESIGN

Initial process modeling captures the business process flow details at the Process and Stage Flow level.

XSOL uses its propriety Enterprise System Language (ESL) Business Process language to enable the description of all process elements. The process capture is undertaken by business analysts and business users through a combination of workshops and access to process documentation. Design information is entered directly into XSOL modeling, creating the relevant process elements in a visual framework. As each element is added the XML based structure of the process model is built up creating an ESL model file.

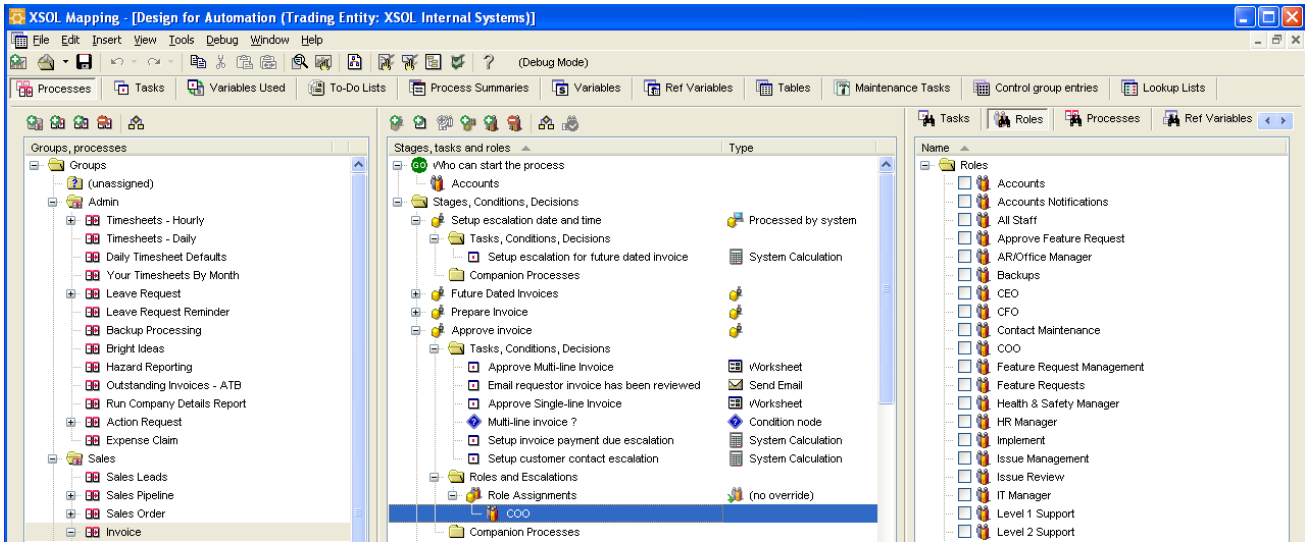


Above: The Process Flow window showing Process Flow with the selected Stage Flow for the 'Approve Invoice' Stage visible below.

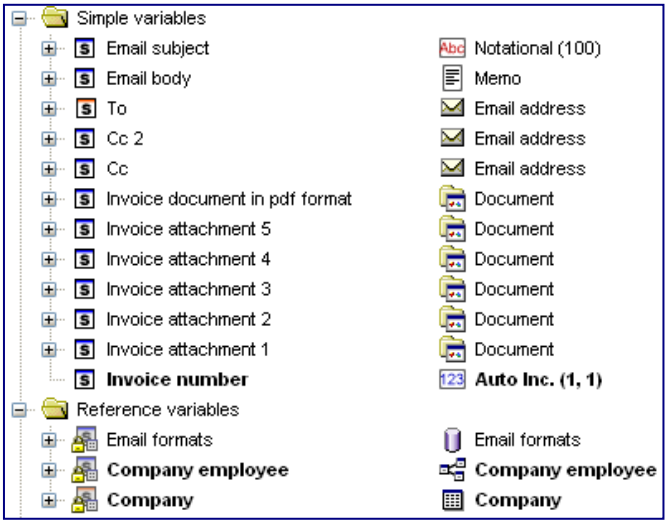
The ESL process language allows for the interrogation of the model for structural integrity. Typically process design goes through several phases beginning with the kind of simplistic process flow typically produced for documentation only purposes. Gradually as more detail is added (particularly in the form of system actions (such as calculation tasks, database updates etc.) the flow becomes more complex and begins to move away from that which would be easily understandable by a normal business user.

Responsibilities, in the form of Roles, are added to Process and Stage elements to create a basis for the creation of a To-Do List for individual users which may contain a variety of Roles which can at runtime be added or removed (as well as associated with Teams and Groups).

Below: The Tree Designer view of the Process and Stage Flow with a list of existing Roles visible on the right

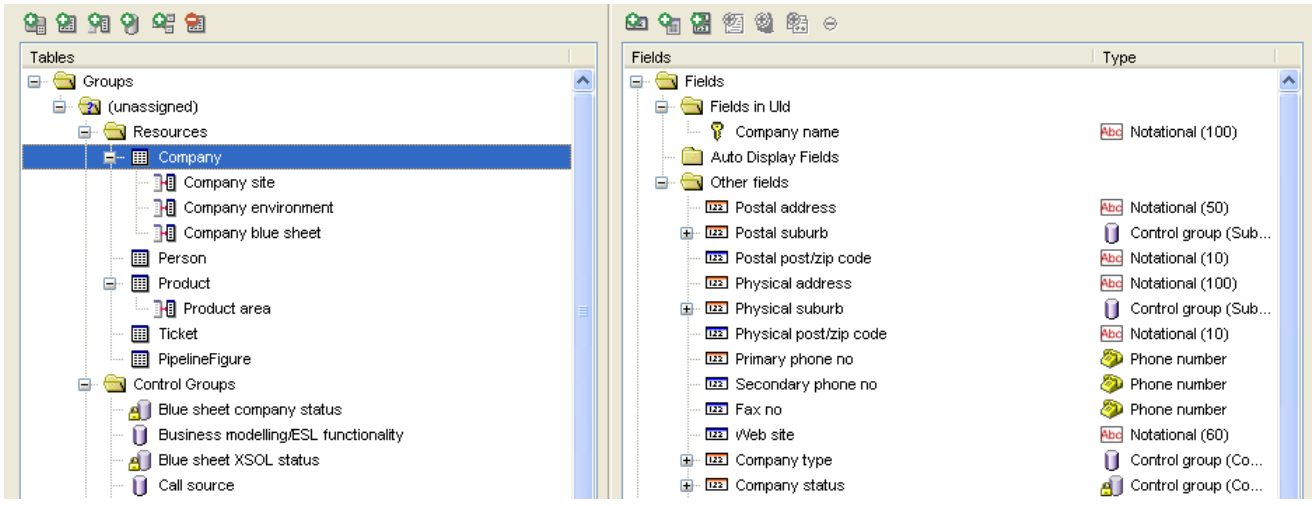


Right: A selection of Simple and Reference Variables assigned to a Task



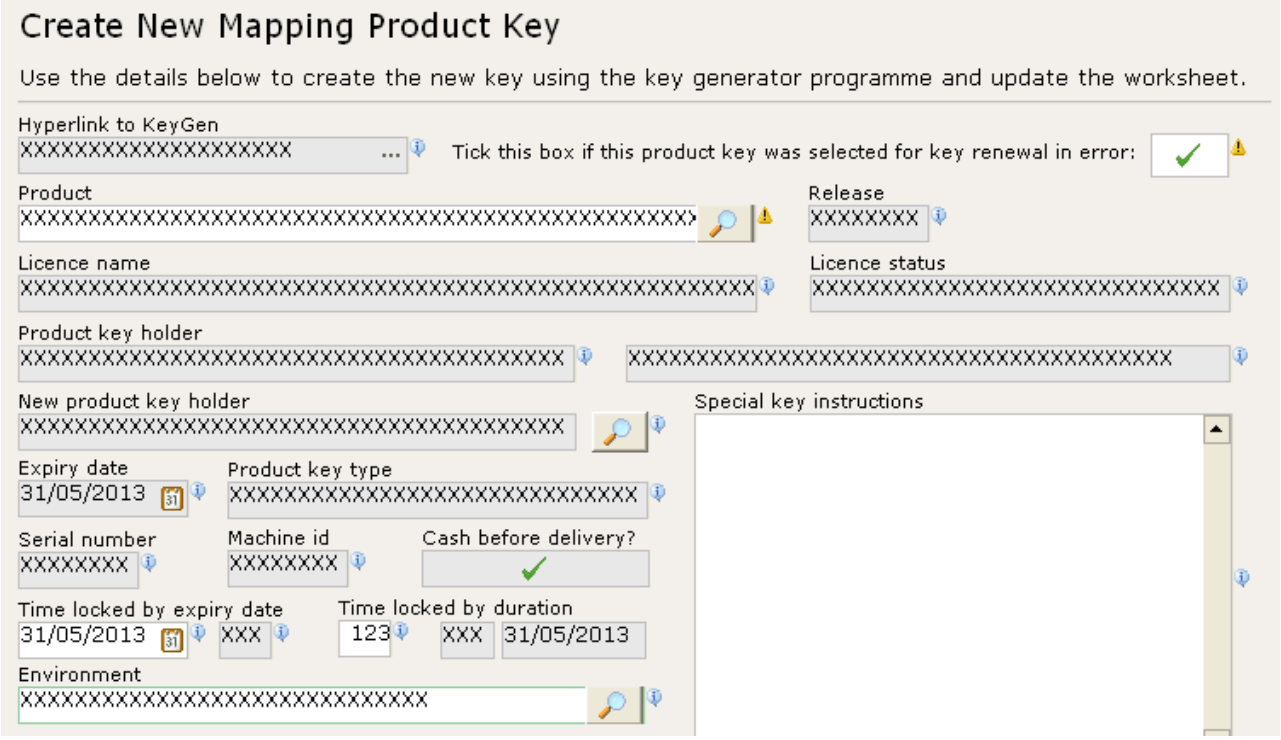
Once the flow is designed the Tasks within a Stage, which represent work undertaken by a user at their desk, have variables associated with them. Variables collect or display information within a process. They can contain data for varying lengths of time depending on their type. A wide range of variable types are available to enable the capture of different information such as text strings, dates, numbers, binaries etc. Variables may be stored only within a process or display information stored in separate database tables (reference variables). The latter allows information to be stored and reused throughout the entire process model and runtime system. If Reference Variables are required then SQL relational database tables need to be created in the model allowing tables (and their data fields) to be used as Reference variables.

Below: A selection of user defined tables and the data fields for the Company Resource



Worksheet Tasks provide forms for a user to interact with in the runtime system. Variables assigned to Tasks can be dropped onto worksheets and arranged. The appearance of the variables can be altered as can their purpose i.e. whether they are mandatory, read-only etc.

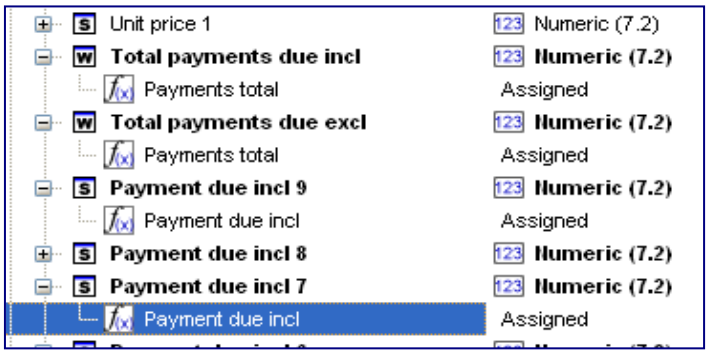
Below: An example of a worksheet in the XSOL Designer with variables (both simple and reference) displayed.



Variables that are calculated have Expressions and Business rules associated with them. These expressions and rules are assigned at specific points when they are required to be calculated.

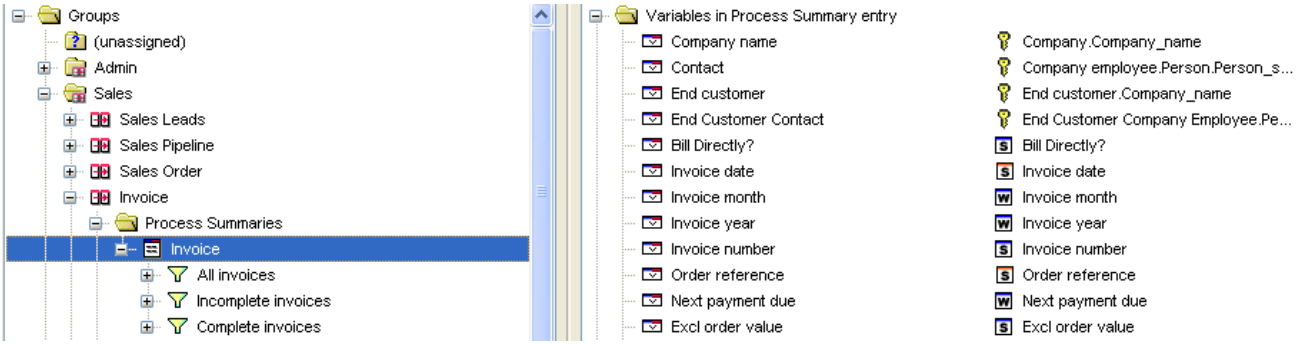
Document repositories are specified as abstract entities and assigned to real document locations (network folders etc.) at runtime.

Once a process is defined a Process Summary can be created for it which at runtime provides users with a dashboard of process movement, progress and visibility of work undertaken and values added to a process instance. Filters can be assigned to a summary limiting the data brought back.



Above: Numeric Variables with associated Assigned Expressions

Below: Process Summary values for an Invoice summary with three Filters



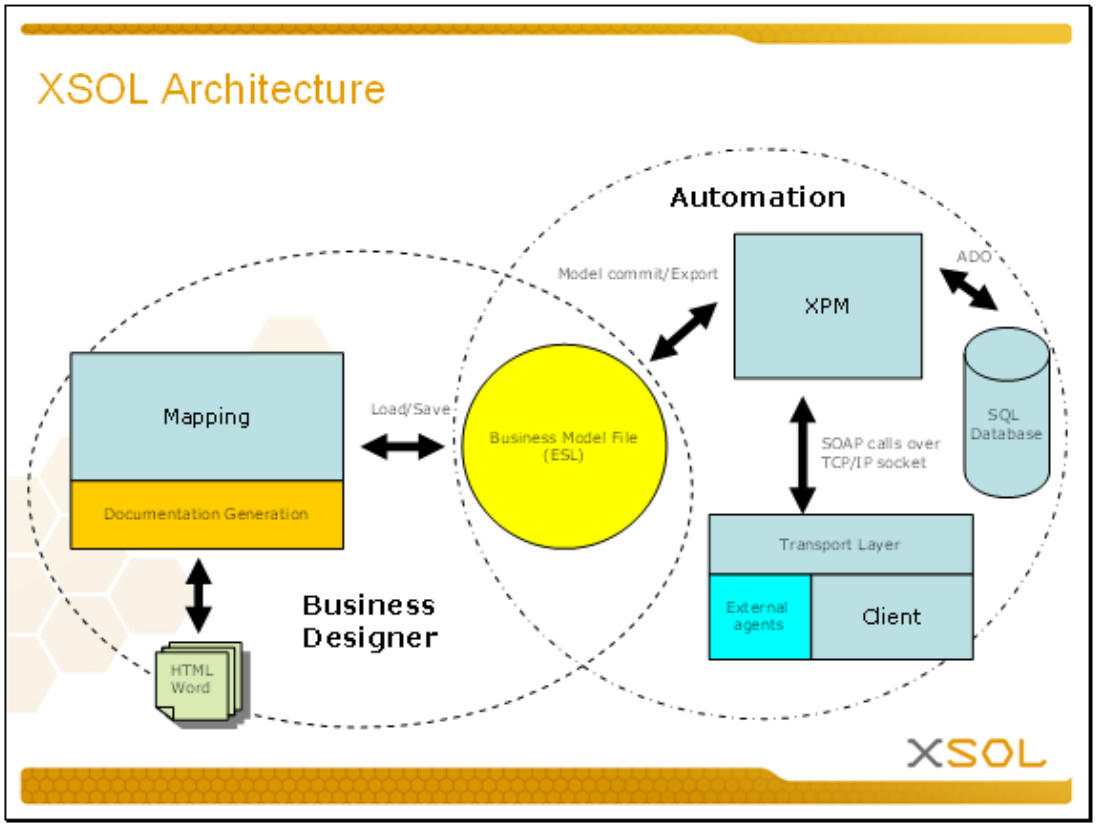
CREATING A RUNTIME SYSTEM

Once the model is structurally complete, determined by an automated check of all the entities within the file, it can be committed to the XSOL InControl Automation system. The base **XSOL InControl Automation** product consists of two products: the Server and Client. The Server has the following components:

- the XSOL Process Manager (XPM)
- a blank SQL Server Database (named by default XPMII)
- XSOL Security Administrator
- XSOL Automation Setup Manager
- XSOL Email Agent (optional or could run on network server)

The XSOL InControl Automation Client is a desktop application installed on all users' desktops or Citrix. Once designed the ESL process model is used directly to fully generate the actual XSOL Automation system in SQL Server, through the XSOL InControl Automation Client, by populating the chosen database. This is called **committing** the model.

XSOL Automation has a three-tier architecture which works in the following way. Win32 Clients connect to the database via the XPM over a TCP socket. All access to the database is via the XPM connection so no direct access is available to the client or to SQL Server. In fact all external interactions such as access to email, documents, printers etc. is carried out via the XPM so all access to these can be strictly controlled.



NOTE: The XPM can be run on the same server as SQL Server or a dedicated server.

Once a system is committed physical users are created (or imported via LDAP) and associated with the Roles specified in the model file via the XSOL Security Administrator. Abstract Document and Printer Locations also specified within the model file are associated with real networked document folders and printers using the XSOL Automation Setup Manager. If Emails need to be sent as part of the system then this is done via the XSOL Email Agent.