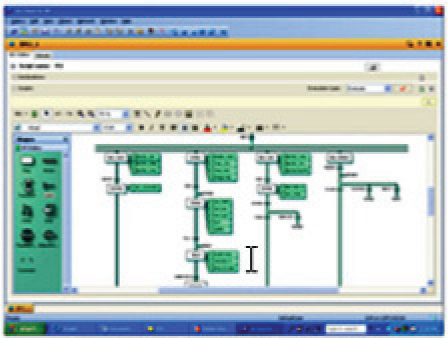
PRODUCT SPECIFICATIONS

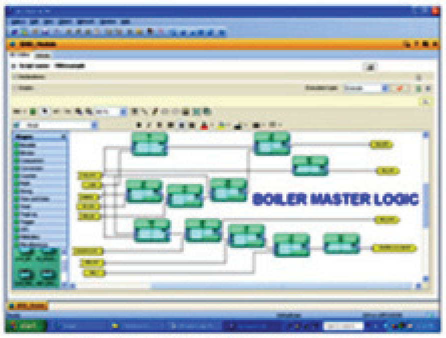
InFusion™ Software

PSS 21S-10C1 B3

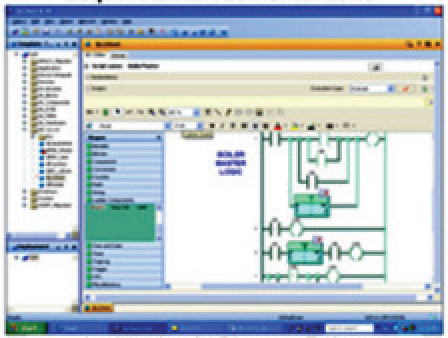
InFusion™ Logic Module 61131-3



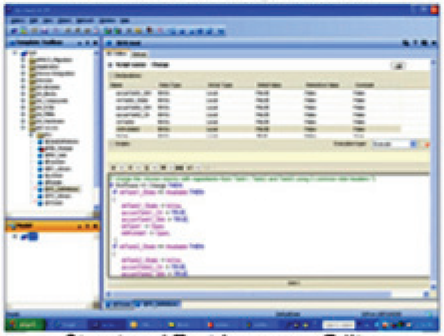
Sequential Function Chart Editor



Function Block Diagram Editor



Ladder Logic Diagram Editor



Structured Text Language Editor

The InFusion Logic Module (ILM) is an engineering tool that creates fully compliant IEC 61131-3 Application Objects that run on InFusion Application Environment (IAE) platforms. Program Modules and Function Block Objects can be created using any of the ILM editors.

FEATURES

InFusion Logic Module enables for a user:

- Graphical construction and deployment of a network of IEC 61131-3 Programs.
- Construction of re-usable Function Block and Module objects to be used in the creation of IEC 61131-3 Programs.
- Ability to Import and Export ILM Application Objects.
- Ability to create and deploy Application Objects that can interchange information with I/A Series and other automation systems such as PLCs.
- Ability to create and deploy Application Objects that can interchange information with any application supporting OPC.

invensys®

OVERVIEW

The InFusion Logic Module (ILM) is an engineering tool that creates fully compliant IEC 61131 Application Objects that run on InFusion Application Environment (IAE) platforms. Program Modules, Function Block Objects can be created using any of the ILM editors. These are:

- Sequential Function Chart (SFC)
- Function Block Diagram (FBD)
- Ladder Logic Diagram (LLD)
- Structured Text Language (STL)

The Application Objects created by this engineering tool are independently scheduled programs that interact with each other and other applications like I/A Series controllers or other automation subsystems such as Programmable Logic Controllers (PLC).

InFusion Logic Modules are constructed from within the InFusion Engineering Environment (IEE) and execute within the InFusion Application Environment. The IEE supports five construction environments. These include:

- The Template Toolbox for constructing re-usable Templates
- The Model View to create and deploy instances of ILM Application Objects
- The Derivation View showing the inheritance structure of derived Templates and Instances
- The Network View to assign ILM Application Objects to I/A Series Workstations
- The Deployment View to assign ILM Application Objects to IAE platforms

The example in Figure 1, shows how a user could create a new derived template from the base $Module template in the Template Toolbox.

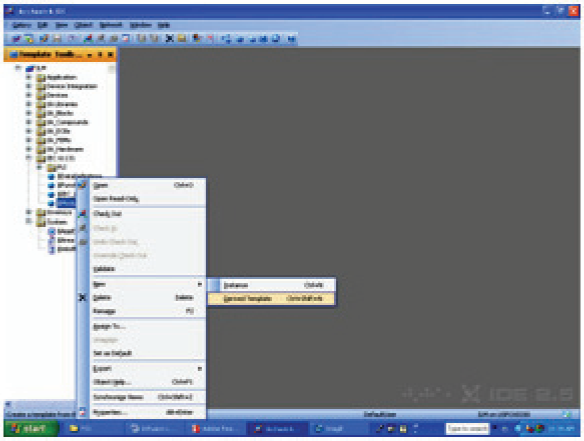


Figure 1. Creation of New derived template

Once a derived template or instance is created and optionally renamed, its editor can be invoked and scripts can be created to establish its desired behavior. Each ILM object can support one Module and as many functions and/or function blocks as required. The example as shown in Figure 2, shows the editor for our newly created ILM object.

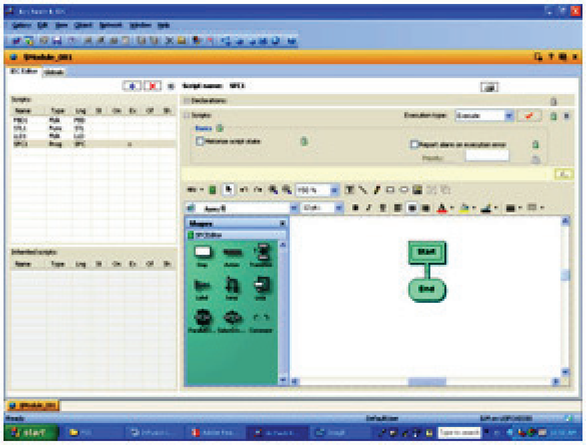


Figure 2. Editor for newly created ILM object

In this example, the user has created four scripts by selecting the + icon above the Scripts pane. One script was created in each of the supported IEC 61131-3 editors. The type and language of each script is also shown in the scripts pane. The currently opened SFC1 script is the program or top level script for this ILM object. It will use (reference) all of the other scripts either directly or indirectly.

To construct an SFC Network within the SFC editor, as shown in Figure 3, the user would simply drag the various network items from the stencil onto the canvas and drop it onto the appropriate place on the network drawing.

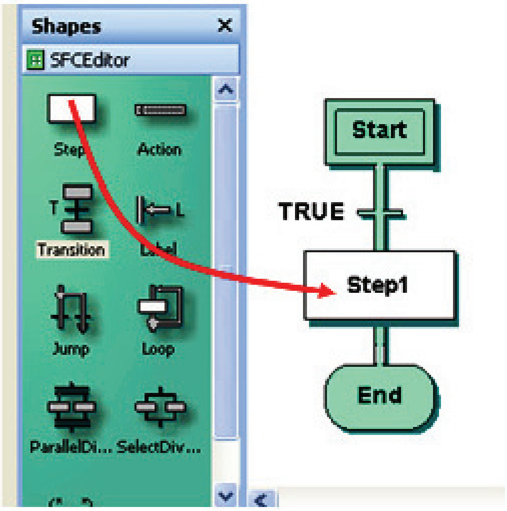


Figure 3. SFC Editor

To directly reference another script, the user would select the f(x) icon on the canvas. In the case of SFC networks, functions and function blocks can only be referenced from within the Steps and Transitions within the network. The user would double-click on one of these to invoke its respective editor. In our example as shown in Figure 3, the user would double-click on Step1 to invoke its editor and then click on the f(x) icon to produce the following dialog as shown in Figure 4.

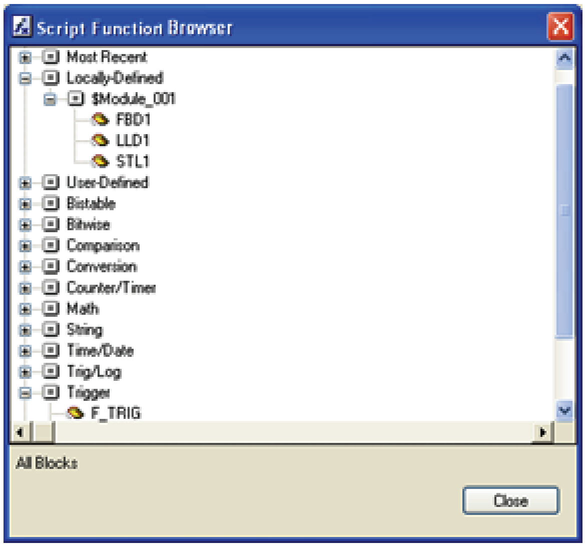
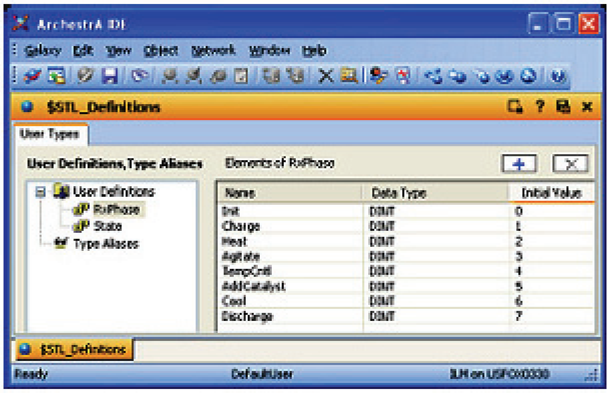


Figure 4. Script Function Browser

Notice the Script Function Browser dialog shows the three function/function blocks under the Locally-Defined/$Module_001 node. These are the scripts we created as part of this ILM object.

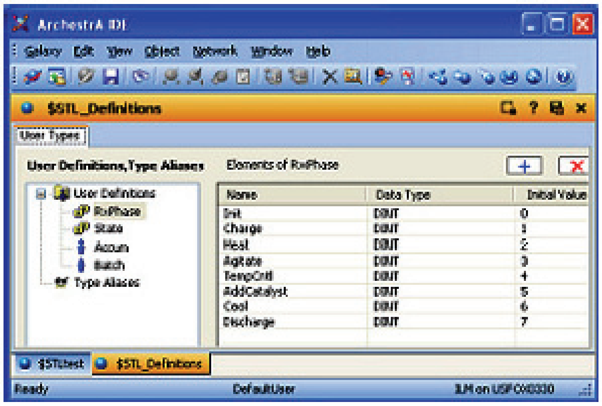
Other categories include User-Defined which would be user defined Function Block Object templates as well as the wide variety of predefined functions and function blocks specified as part of the IEC 61131-3 requirements.

The ILM engineering tool also enables the user to construct complex data types such as structures and enumeration types. This is done by deriving from the $DataDefinitions template. Figure 5 and Figure 6 show examples of enumeration and structured data types.



Name	Data Type	Initial Value
Init	DINT	0
Charge	DINT	1
Heat	DINT	2
Agitate	DINT	3
TempCntl	DINT	4
AddCatalyst	DINT	5
Cool	DINT	6
Discharge	DINT	7

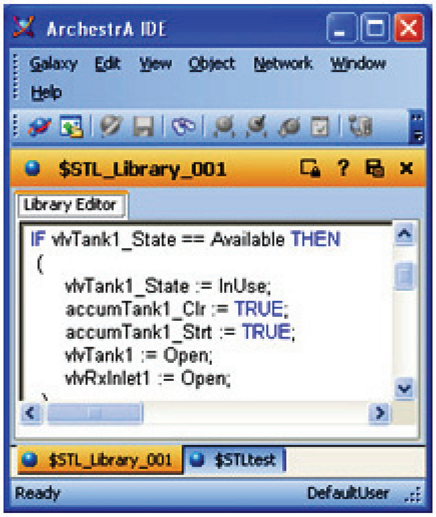
Figure 5. Example of enumeration



Name	Data Type	Initial Value
Init	DINT	0
Charge	DINT	1
Heat	DINT	2
Agitate	DINT	3
TempCntl	DINT	4
AddCatalyst	DINT	5
Cool	DINT	6
Discharge	DINT	7

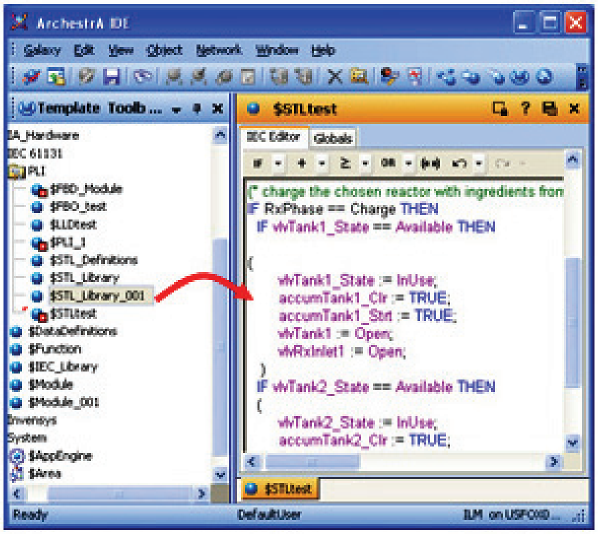
Figure 6. Example of structure type

The ILM engineering tool also enables the user to construct snippets or code fragments and store them as Library Objects by first deriving an object template from $IEC_Library and then storing the code fragment in this Object. In the example shown in Figure 7, a fragment of code is stored in $STL_Library_001 and then copied into $STLtest as shown in Figure 8.



```
IF vlvTank1_State == Available THEN
(
    vlvTank1_State := InUse;
    accumTank1_Clr := TRUE;
    accumTank1_Strt := TRUE;
    vlvTank1 := Open;
    vlvRxInlet1 := Open;
```

Figure 7. Fragment of code stored in $STL_Library_001



```
(* charge the chosen reactor with ingredients fron
IF RxPhase == Charge THEN
  IF vlvTank1_State == Available THEN

(
     vlvTank1_State := InUse;
     accumTank1_Clr := TRUE;
     accumTank1_Strt := TRUE;
     vlvTank1 := Open;
     vlvRxInlet1 := Open;
  )
  IF vlvTank2_State == Available THEN
  (
     vlvTank2_State := InUse;
     accumTank2_Clr := TRUE;
```

Figure 8. Fragment of code copied into $STLtest

Instances of user created ILM Modules can be deployed into any IAE platform creating a network of interacting Application Objects that also communicate with applications mapped to the IAE Device Integration Objects. These include I/A Series, a wide variety of PLC systems, Enterprise Management software including Asset Management and Supply Chain.

In the example as shown in Figure 9 below, IEE is used to create a deployable set of instances consisting of a Platform Object, Application Engine Object and Area Object. These are sent to a remote computer to create the environment to receive our ILM Object. Once these are installed on the remote platform, our ILM Application Object can be deployed.

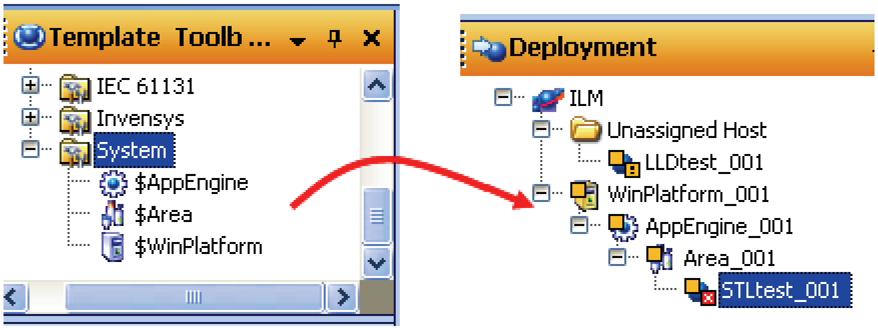


Figure 9. ILM Application Object Deployed

HARDWARE AND SOFTWARE PREREQUISITES

InFusion Engineering Environment Server

- I/A Series Station type: P91
- Operating System: Microsoft® Windows 2003 Server Software
- Microsoft .NET Framework 1.1
- InFusion Engineering Application Environment
- Computer: Intel® Pentium® 4, 1.8 GHz (or higher) processor
- Memory: 2.0 GB of main memory
- Hard Disk: Recommended: 16.0 GB free memory
- Video Graphic Accelerator Card: 32 MB of memory
- Communications Network: 100 MHz TCP/IP Switched Ethernet

InFusion Engineering Environment Client (may be on same platform as Server)

- I/A Series Station type: P91 or P92
- Operating System: Microsoft® Windows XP or Windows 2003
- Microsoft .NET Framework 1.1
- InFusion Engineering Environment Client Software
- Computer: Intel® Pentium® 4, 1.8 GHz (or higher) processor
- Memory: 2.0 GB of main memory
- Hard Disk: Recommended: 16.0 GB free memory
- Video Graphic Accelerator Card: 32 MB of memory
- Communications Network: 100 MHz TCP/IP Switched Ethernet

invensys®

33 Commercial Street
Foxboro, MA 02035-2099
United States of America
www.foxboro.com
Inside U.S.: 1-866-746-6477
Outside U.S.: 1-508-549-2424
or contact your local Invensys
representative.
Facsimile: 1-508-549-4999