Symphony Plus
S+ Engineering: Composer Harmony
Symphony™ Plus includes a comprehensive suite of engineering tools. S+ Engineering’s Composer tools provide a visual environment for easy configuration of control system strategies, global configuration databases, management of system libraries of reusable software components, and integration and management of intelligent field and electrical devices.

The working environment provided by Composer simplifies the configuration and maintenance of Symphony Harmony and Symphony Plus systems. Composer is designed to operate on Microsoft Windows 7 Professional or Server 2008 and support both 32-bit and 64-bit versions. It is compatible with INFI 90 OPEN system configurations and is capable of importing existing WinTools configurations. Once imported, these configurations can be fully integrated into Composer and use all its features.

Additional features

Configuration viewing and monitoring
View and Monitor is an optional Composer add-on feature that allows control logic documents (CLDs) to be monitored from a Web browser in read-only mode. View and Monitor functionality is based on Microsoft’s Internet Information Server (IIS) technology. Using the base Composer functionality, CLDs can be published in SVG format in a virtual directory on an IIS workstation. The IIS workstation provides the contents of virtual directories as websites to any Web browser.

Multiuser client/server architecture
Composer applications use client/server technology to support multiple users operating in a networked environment. Configuration information managed by Composer’s configuration server can be accessed simultaneously by multiple users. In addition to interacting with configuration information, users can access online data from a running Symphony Plus, Symphony or INFI 90 OPEN system by using Composer’s communication server. The Composer architecture supports one configuration server per system and multiple communication servers.

Configuration database
A configuration server can support up to 10 simultaneous client connections and provide users with shared access to a system’s configuration information. Composer’s configuration server manages and stores configuration data in one configuration database per project or system. This configuration database eliminates duplication of data entries, simplifies database management, and automates many configuration tasks. Information can be imported and exported in many formats.

Object exchange
Composer introduces a single, system-wide component database called object exchange. Object exchange provides a multiuser repository for all standard symbols, macros, control strategies and control logic templates used to generate control system strategies. Object exchange is a prominent part of Composer client applications. Users are encouraged to use system and project standards when creating automation strategies. In addition to presenting standard system objects, object exchange enables users to add components they have created in object exchange.

Intelligent field devices integration
Composer fully supports configuration and maintenance of field and electrical devices connected to the HPC800 controller via PROFIBUS and HART communication protocols. Each device’s resident information is available for use in the HPC800’s control logic using Composer and standard function codes. Also, within the S+ Engineering tool suite, Composer Field is used to easily configure, maintain and manage intelligent devices through a user friendly graphical environment via FDT/DTM technologies.

Custom C programs
Composer’s base client provides the ability to load custom C programs without the need for an additional utility. Examples of custom applications include foreign device interfaces and performance calculations.

Batch data manager
Batch data manager (BDM) is a family of engineering tools for creating, editing, managing, downloading and debugging batch, sequential, and user-defined function (UDF) code configurations. BDM supersedes and provides migration from all previous batch and UDF tools. Batch 90 for batch sequencing and UDF codes is discussed in more detail in the optional client applications section of this overview.
Composer applications

The base Composer product contains all the functionality necessary to develop and maintain Symphony Plus control system configurations. There are two primary applications: explorer and automation architect.

Explorer

The primary application of Composer is the explorer. Explorer presents the Symphony Plus, Symphony or INFI 90 OPEN system architecture and provides an intuitive means for organizing, navigating and locating system configuration information. Explorer presents a user with two main windows: system architecture and object exchange.

System windows

The system architecture window functions similarly to Microsoft’s file explorer. The left pane of the window displays a hierarchical representation of the Symphony Plus system. When a system object is selected, the right pane displays a detailed view for the selected object.

The system window supports two views: the document view (figure 1) and the data browser view (figure 2). When the system window is in the document view, it will show the configuration documents that are associated with the system object that the user has selected. Configuration documents support long file names and can include control logic documents, human system interface displays, or documents created by other applications such as CAD packages or spreadsheets.

The ability to associate any documents with the system architecture is an important feature. This allows any information, such as piping and instrumentation diagrams (P&IDs), cabinet arrangement drawings or field wiring drawings, to be managed by the configuration server and accessed by Composer client applications. All that is required to edit any of these documents is to double-click the document. Composer’s explorer will automatically launch the appropriate application for the document selected.

When the system window is in the data browser view, the right pane of the system architecture window will display tag information associated with the system object the user has selected. All tag information presented is retrieved from the configuration server database that is managed by the Composer server.

When working in the data browser view, users can view, define and modify tag data for the whole system. This central repository of data is managed by Composer’s configuration server for all tag data in the entire system. The data for each tag is added to the configuration server database as each tag is defined. This eliminates the need for users to enter the same information more than once. Some notable features of the data browser view are the ability to:
- Edit tag objects in a datasheet or property page view
- Filter the database: filtering makes configuration easier and faster by eliminating unnecessary information from view
- Import and export tag data
- Navigate directly from a tag to its related configuration document
- Perform automatic search and replace operations based on complex queries
Users define their own folders within object exchange.

User-defined folders can be nested as deeply as the user desires and provide a container for user-defined objects. User-defined objects can include macro logic, shapes, symbols, standard drawings or control logic templates (CLT). In addition, user-defined folders can contain references or shortcuts to system-defined objects. This enables users to effectively organize standard system objects to best suit their needs.

To support reuse, object exchange provides library management features such as cutting, copying and pasting objects between different projects. This makes it easy for system engineers to share objects among projects.

**Automation architect**

Automation architect provides for visual creation, editing, monitoring and tuning of control logic. High-level control strategies can be created by dragging and dropping function codes from object exchange to the control logic document.
Control strategies are represented graphically by automation architect. Rather than textually programming strategies, automation architect represents pre-defined control strategies as function blocks. By connecting function blocks (figure 4) users are able to specify the signal flow of a control strategy and visually define the control strategy.

Automation architect stores configuration information in control logic documents. Control logic documents support grouping of multiple logic sheets in a single document. This allows users to group sheets of logic together using process partitions. For example, a single control logic document could be used to define the control strategy for a boiler. Each control loop or interlock sheet associated with the boiler could be assigned to the control logic document. Partitioning control logic in this manner is more process object-oriented and intuitive to process engineering personnel (figure 5).

These values are automatically presented on the same control logic documents that were used to configure the module. Composer’s tuning functionality allows the change of logic parameters as allowed by the controller. The control logic document in the Composer application and in the module are dynamically updated when tuning changes are made. This means that documentation for the system accurately reflects the current configuration of the controller.

Control logic templates
Two of the primary goals of Composer are to reduce the cost of implementing control strategies and to improve the quality of control strategy software. To realize these goals, Composer supports a new type of document known as a control logic template.
Control logic templates (figure 7) define reusable standard control strategies that are typically used to develop a process automation system. They can be thought of as blueprints that define the structure of a control strategy. They are maintained by object exchange and can be used to quickly define control logic documents.

The control logic template linking functionality allows users to define logic that is controlled by the template or that can be modified on each configured instance. Any subsequent changes can then be spread to all linked instances. When a template updates its linked instances, it will preserve instance-specific configuration. This template management functionality provides efficient maintenance and utilization of reusable standard control logic.

Intelligent field devices configuration and management
Within the S+ Engineering tool suite, Composer Field fully supports configuration, commissioning, maintenance, and management of intelligent field and electrical devices using PROFIBUS and HART communication protocols (e.g., transmitters, actuators, motor control centers (MCC), etc). The data or variables inside intelligent devices can be accessed by function blocks in the HPC800 controller, in a similar way as traditional I/O channels data is accessed. An engineer can use Composer Automation Architect to configure data from intelligent devices and from I/O channels equally for real time control strategies.

Beyond function block data access, Composer Field’s field device tool (FDT) provides a graphical environment to configure and manage intelligent devices using device type manager (DTM) technology, in a similar way of configuring a printer by its driver in Windows. User can use the DTM to check basic information of device, set device parameters and characteristics, change device internal calculation mechanism, do simulation or forcing for testing, etc. All of this can be done in real time and within a user friendly graphical presentation environment.

For conventional device description files (GSD), a basic PROFIBUS DTM is available to allow standardized offline configuration. HART devices are configured and parameterized via standard HART protocols without the need for additional tools by using a standard HART DTM. The individual PROFIBUS or HART DTMs can be accessed from Composer Field’s multiple data views, such as the system or location overview and others. Composer Field also includes automatic net calculation and loading of process items by using the device-specific channel configuration generated from the DTM.
Optional client applications

Bulk engineering
All the controller configuration and tag configuration information is available via the object linking and embedding (OLE) interface client option of Composer. Using the OLE interface, users can access, extract and replace configuration information stored in the Composer configuration server. Any application that is capable of communicating via OLE/COM can access information from the Composer configuration server via the OLE interface. An example of an application that is capable of accessing information via OLE is Microsoft Excel.

The primary advantage of accessing data via Composer’s OLE interface is that it allows users to extend the capabilities of Composer with custom scripts or add-on applications. At the same time, users are able to rely upon the extensive data concordance rules provided by the Composer configuration server. This provides the user with the capability for performing bulk configuration editing in Excel and importing it into Composer’s configuration server.

Batch data manager
Batch data manager (BDM) is a family of engineering tools for creating, editing, managing, downloading and debugging batch, sequential and user-defined function code configurations.

The heart of BDM is the Batch 90 programming language. Batch 90 enables the engineer to create batch and sequential control applications using clear, concise, natural syntax control statements. Batch 90 is layered on top of regulatory and discrete device control function code logic. Batch 90 programs can be used to change controller set points, turn discrete devices on and off, change modes and perform a host of other supervisory operations. In effect, function codes execute base regulatory and discrete device control while the Batch 90 program performs supervisory control and process operation.

A Batch 90 program comprises all the phases that can be run on a specific class of equipment. A phase is the smallest element of control that can accomplish a process-oriented task on a specific class of equipment.

A Batch 90 phase provides a structured, modular approach to batch design by including distinct logic sections for:

- Normal logic, which executes standard sequential control
- Continuous control logic, which continuously executes a set of interlocks
- Fault logic, which executes if a stop command is issued or if a fault is detected in normal logic or continuous control logic
- Hold logic, which executes if a hold command is issued
- Restart logic, which executes return-to-normal logic from the held or stopped states

Batch 90 programs are capable of executing up to 32 phases in parallel. A Batch 90 program can be written using generic function block references so that only one program for a class of process equipment needs to be written. This methodology is often called ‘unit-relative programming’ and it helps to minimize engineering effort. The common sequence or CSEQ function block and data structure within the Batch 90 programming language make it possible for Batch 90 programs to establish a client/server relationship. This enables Batch 90 applications to synchronize and/or supervise each other, regardless of where (ie, in which PCU or controller) the Batch 90 programs are running.

Recipes are used to determine the procedure and formulation of the batch. A procedure determines the order in which the Batch 90 phases are executed. Formulation is the set of data that defines the requirements for a specific product. For example, formulation may include target flow rates, quantities, temperatures and times.

BDM applications reference their configurations in projects. This is similar to the Composer concept of a project, but BDM projects are managed separately. A BDM project can be located on the computer local to the user, or it can be accessed remotely over a network to a mapped drive if a file server is used. A project typically encompasses a process area. A user-defined function (UDF) project includes the required UDF programs for a process area. A batch project references a class library that includes a class for each type of equipment in a process area. Each class contains the Batch 90 program associated with the class of equipment plus the unit definition files if unit relative programming is implemented. If the unit procedure editor or master recipe editor is used, recipes are created and managed in a separate window of the same project.
BDM comprises:

**Batch tools for batch sequencing**

Batch tools for batch sequencing include a full-featured text editor to create and compile Batch 90 programs that are downloaded to the batch sequence (BSEQ) function code (figure 10). Also included are the unit procedure editor (figure 11) and the master recipe editor, which are used respectively to create unit procedures and master recipes in a graphical format. Creation of master recipes is optional; they provide a method for linking unit procedures for a specific production train. In this batch execution model, recipes are created using BDM and downloaded to the controller in their entirety before the batch begins. This execution model provides high system integrity, as all the information needed to execute the batch is self-contained in the controller. It is especially useful when a limited number of products is produced or for sequence control applications where the sequence is rarely modified.

**User-defined function codes**

User-defined function (UDF) codes includes a full-featured text editor to create and compile UDF programs using natural syntax control statements similar to Batch 90. UDF programs can be downloaded to UDF-specific function codes (figure 12) and incorporated into control strategies in the same manner as any other function code. The UDF program does not require a recipe or an operator’s input; it begins executing when the controller begins executing. This product is especially useful for users who need to create function codes unique to their process, or for sequence control applications that execute continuously and do not require a recipe.
The dynamic debugger, the primary troubleshooting tool for Batch 90 and UDF programs, is used with programming languages. The dynamic debugger (figure 13) enables engineers to:

- Observe process data such as variables and set points, as well as intermediate computational values from the program including timers, ramps and integrators
- Override normal operation of pumps, valves and control loops, and direct their activities from the debugger
- Stop normal program operation and single-step the program to analyze data without affecting other programs or function block execution
- Assign breakpoints. The program can be designed to run to a certain point and then stop execution, allowing engineers to systematically analyze logic
- Perform online recipe parameter editing when used with the unit procedure editor or the master recipe editor

System architectures
Composer software consists of three main components; configuration server, communications server and client applications. The configuration server software manages the INFI 90 OPEN, Symphony or Symphony Plus system definition within a Composer project database. System definition includes specification of the system architecture according to: control networks, console nodes, process control unit or PCU nodes and controllers. All composer system definition files, along with linked third-party documents (ie, Excel spreadsheets, AutoCAD drawings, etc), are managed within the Composer project database. The communications server software connects to the Plant Network (PN800) via a Plant Network Interface (PNI) or the INFI-NET control network via an INFI-NET to Computer Interface (ICI). The communications server manages the physical connection to a PNI or ICI and provides Composer client applications with access to Composer runtime communications. Details of the physical connection (ie, PNI or ICI type, Ethernet TCP/IP address, etc) are defined by Logical ICI in Harmony System Configuration Utility. Composer client applications such as Monitor/Tune (Automation Architect), or Save/Load/Verify (Explorer), and other functions access runtime communications by selecting an appropriate Logical ICI from the list of up to 20 total Logical ICIs in the system.
Composer supports INFI 90 OPEN, Symphony and Symphony Plus systems of all sizes. For small systems, as shown in figure 14 and 15, Composer’s configuration server, communications server and client applications are all loaded onto a single engineering workstation PC. The software components communicate with each other using inter-process communications within the engineering workstation PC.

For larger systems, Composer’s software supports client/server architecture over an Ethernet TCP/IP network. Figure 15 and 17 shows how the Composer software components may be distributed over a client/server network to provide a multiuser engineering environment. The configuration server (project database) can be loaded on any file server within the Ethernet network that client workstations can access.
The project database is critical to the maintenance of the INFI 90 OPEN, Symphony or Symphony Plus system, and it should therefore be backed-up routinely.

Communications servers support up to 10 client workstation connections. When using Plant Network (PN800) as the system control network, all Composer server and client PCs are directly connected to Ethernet Plant Network, optionally, all in redundant connection. A PNI800 module needs to be connected to the same network segment to handle all data communication, integrity check, security check, etc. When using traditional INFI-NET as the system control network, an Ethernet ICI is connected to INFI-NET by a NIS21 network interface module. It also provides secure Ethernet communication to the Composer server and client PCs via SSL encryption.

International language support
Base Composer applications have been developed to support international languages. Configuration documents developed using Composer applications will accept and preserve user inputs in any language supported by Windows 7 or Windows Server 2008 (figure 18). When a user creates a project, Composer’s configuration server sets the default language for the project to match the default language of the user creating the project. Once the project is created, users can enter text into the documents and database records presented by Composer client applications. While Composer supports any of the languages supported by Windows 7 or Server 2008, users that require international language support should consult their regional ABB office to ensure that the specific products they want to configure support a given language.

Online documentation
Composer provides users with standard Windows help. In addition to this, all Composer documentation is provided in electronic form (figure 19). The instruction manuals for Composer are provided on the Symphony Plus software DVD disk in Adobe PDF format, along with an enhanced version of Adobe Acrobat Reader that supports a sophisticated search engine. The user can right click on any function block in the automation architect window to call up the electronic documentation to find a detailed explanation and configuration examples of the selected Function Code, without leaving the configuration environment.
Contact us

ABB Inc.
Power Generation
Wickliffe, Ohio, USA
Phone: +1 440 585 3087
Email: powergeneration@us.abb.com

ABB AG
Power Generation
Mannheim, Germany
Phone: +49 621 381 3000
Email: powergeneration@de.abb.com

ABB Pte. Ltd.
Power Generation
Singapore
Phone: +65 6776 5711
Email: powergeneration@sg.abb.com

www.abb.com/powergeneration