

Features and Benefits

- **Reduce Engineering Time and Cost:**
Building re-usable custom control applications from a comprehensive set of pre-engineered process control entities will save many hours during project design and implementation.
- **Shorten Project and Startup Schedules:** Leveraging 800xA's object-oriented technology, PCEquipmentLib provides users with the ultimate balance between flexibility and standardization. Standard equipment module types can be defined early in the project and used as the "masters" throughout the plant. PCEquipmentLib provides a toolkit of pre-engineered components and equipment module templates to make this easy and efficient.
- **Maximize Security and Integrity:**
PCEquipmentLib library objects are developed under stringent Quality Assurance standards, factory tested and encapsulated to ensure functional integrity. This can greatly reduce project testing and documentation especially for FDA validated projects.
- **Improve Operator Response to Process and Batch Events:**
PCEquipmentLib searches the control logic and automatically builds dynamic lists of devices, phases, equipment, prompts and timers and creates displays showing the active status of interlocking logic for each entity. Operators can easily see the root cause affecting a phase, an equipment module or a unit - and be assured that what they see matches exactly with the logic executing in the controller.

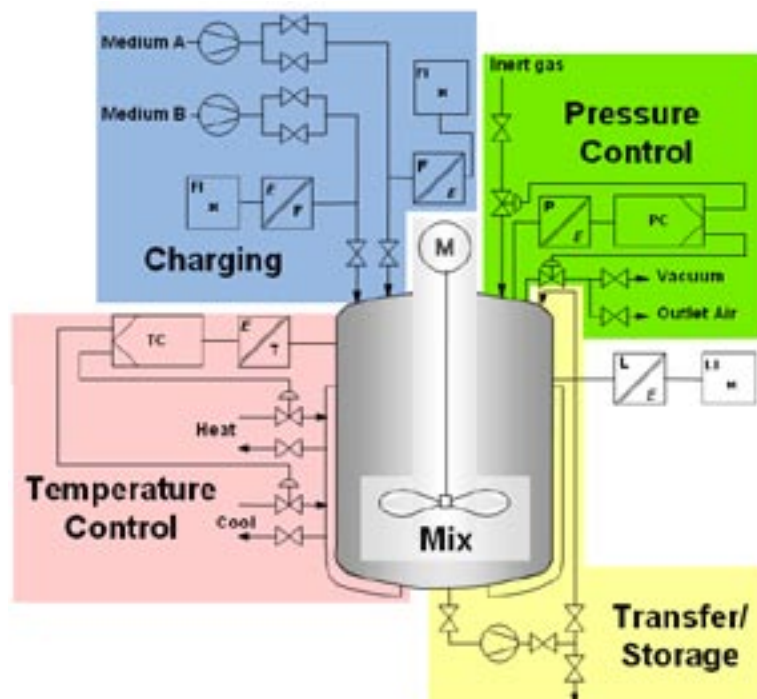


Figure 1. Equipment Modules provide a high level of re-use and consistency in control design

Process Control Equipment Library (PCEquipmentLib) is a comprehensive library of Equipment Module templates, and toolkit components for Industrial IT Extended Automation System 800xA. Part of a broad family of 800xA libraries, PCEquipmentLib is designed to optimize the specification and building of S88 style Equipment Modules and customized Process Units. Designed to "plug and produce" with standard PCDeviceLib control objects, PCEquipmentLib shares common terminology, engineering principles, and naming conventions to make engineering consistent and easy.

Using PCEquipmentLib, automation engineers can create complex customised control applications simply and efficiently, engineered to internationally recognized standards, and totally integrated into the powerful 800xA aspect object architecture. ABB's unique Aspect Objects technology makes the process easy.

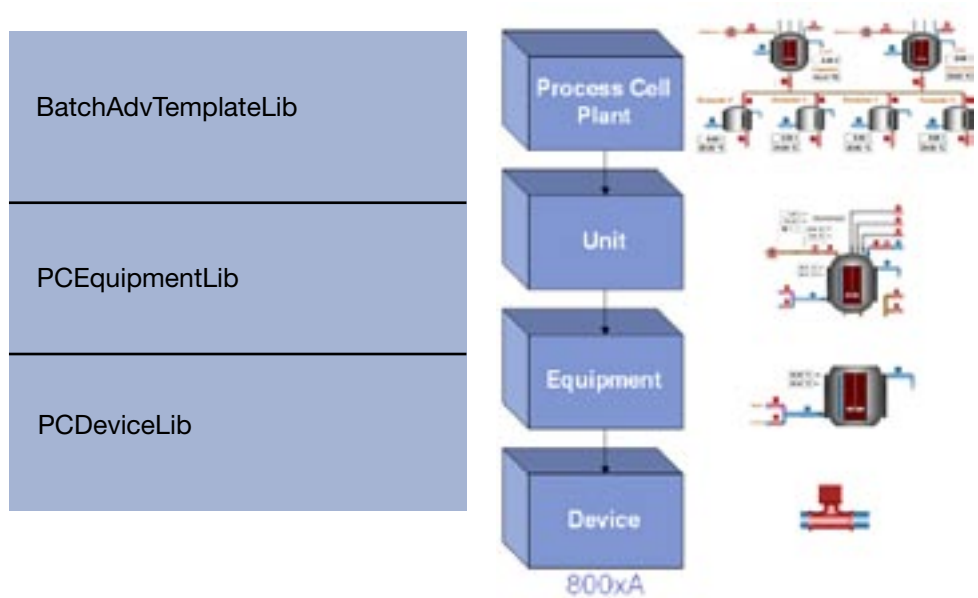


Figure 2. Engineering S88 style Equipment Modules is easy and efficient using PCEquipmentLib

Equipment Modules

Encapsulating devices into Equipment Modules makes the reuse of complex packages of device control simple, accurate and repeatable. Equipment Modules combine together one or more control devices into a common function such as ramp/hold temperature, pull/release vacuum, agitate, add medium, etc. These in turn, can be used across several units to provide a high degree of re-use, ensure operating consistency through the plant and reduce engineering and testing efforts.

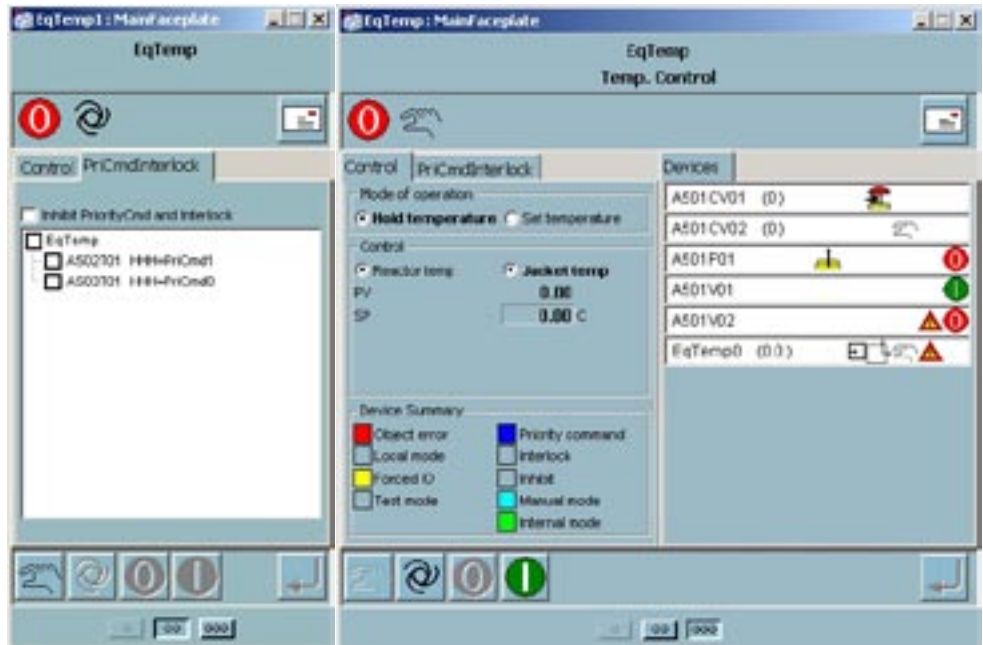


Figure 3. Temperature Control Equipment Module faceplate showing:

- List of devices
- Modes and states
- Priority commands and interlocks

Equipment Modules may utilize simple state engines or complex sequential logic to drive the transitions between different Modes of Operation (MOP).

- The Mode Of Operation can be commanded by different equipment phases, or by an operator from the Equipment Module faceplate under manual control.
- The Mode Of Operation can be activated from different states in the same equipment phase (Running, Holding, Stopping...).
- Different types of process units can use the same equipment module type.

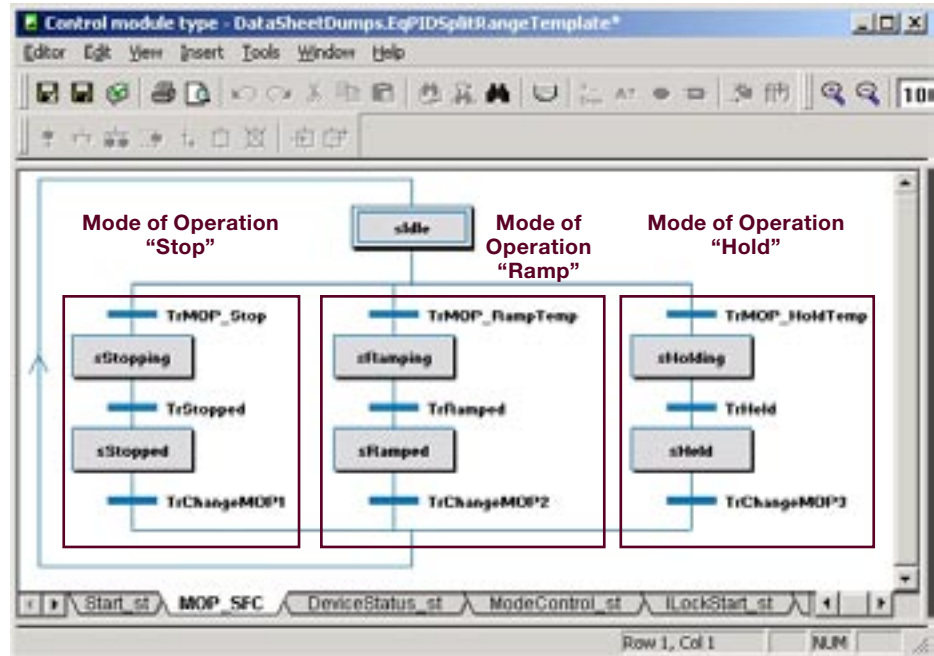


Figure 4. PCEquipmentLib provides an Equipment Module Mode and State Engine, complete with exception handling, packaged into a single control module.

Customized Equipment Modules

Equipment Modules are often highly project specific. In addition to delivering a standard set of Equipment Module templates, PCEquipmentLib provides a toolkit of pre-engineered components to assist with building application specific Equipment Modules.

Toolkit components include:

- Process Timers
- Prompts
- List of Prompts
- List of Timers
- List of Devices
- List of Equipment
- List of Phases
- EQModeControl
- EQDeviceSummary
- EQCore
- PCC Functions for Phases
- PCC Functions for Units

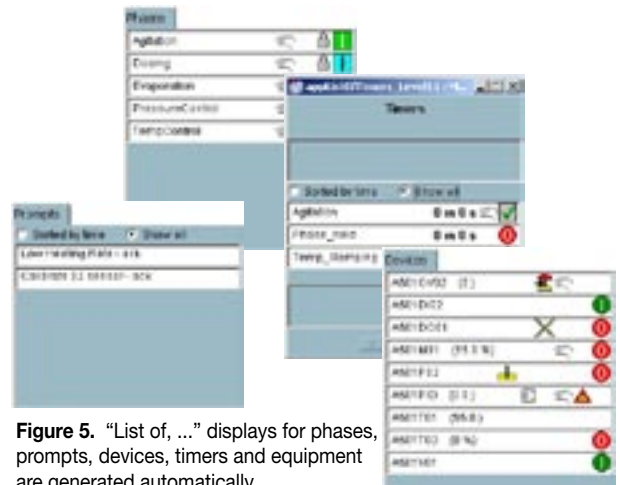


Figure 5. “List of, ...” displays for phases, prompts, devices, timers and equipment are generated automatically

Reuse of pre-engineered and tested PCEquipmentLib components in your custom control logic provides many benefits:

- Lower implementation time and maintenance cost by minimizing documentation and testing efforts
- Common application design for device control and exception handling
- Common HSI on all equipment modules
- Full integration with the 800xA aspect object architecture including access to logical color definitions, Native Language support (NLS), alarm and event handling and the security and audit trail models.

Batch Library Objects

Objects from the 800xA batch library (BatchAdvTemplateLib) such as S88 Process Units and Phases come complete with all the required functionality for most batch applications ‘out of the box’. For the most demanding applications, PCEquipmentLib’s toolkit components can be used to further extend the standard objects with additional functions and faceplate elements.



Figure 6. Standard Phase faceplate

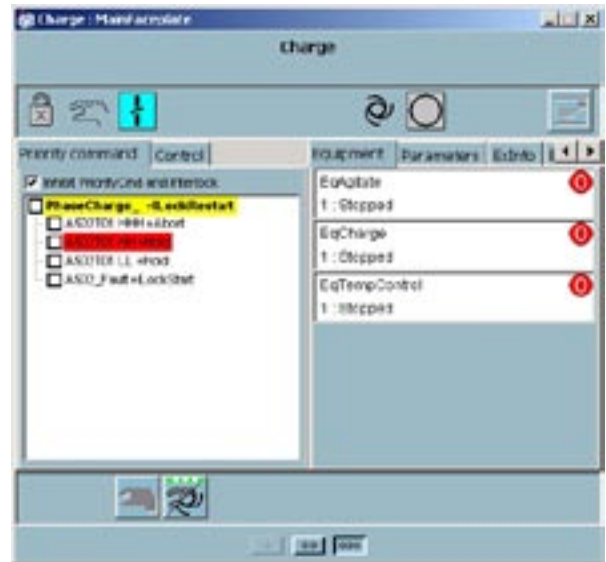


Figure 7. Phase faceplate with added PCC Display (left side) and Equipment List (right side)

Common Exception Handling

In conventional systems, exception handling is configured within the detailed phase application code. This can be complex to implement, and difficult to troubleshoot.

PCEquipmentLib provides an easy mechanism for handling exception logic. By providing a set of functions that the application engineer can use to build control logic, PCEquipmentLib enables the user to graphically:

- Set a Phase or Unit to one of its S88 exception states (PAUSE, HOLD, STOP, ABORT) based upon one or several plant inputs; for example, a high level in a reactor.
- Prevent a Phase or Unit from starting or restarting while one or several plant inputs are in an exception state; for example, a high temperature in a reactor.

Common exception monitoring is achieved without replication of application code across different phases, saving both initial engineering and maintenance effort. Exception monitoring can be independent of the phase application code execution status.

PCEquipmentLib extends the Priority Command (PCC) concept in PCDeviceLib to Phases and Units. The system searches the application for as-built priority commands and interlocks and builds the dynamic elements so that the display always reflects the current controller logic. In this way, presenting the root cause of the problem is consistent and integrated with device level interlock displays.

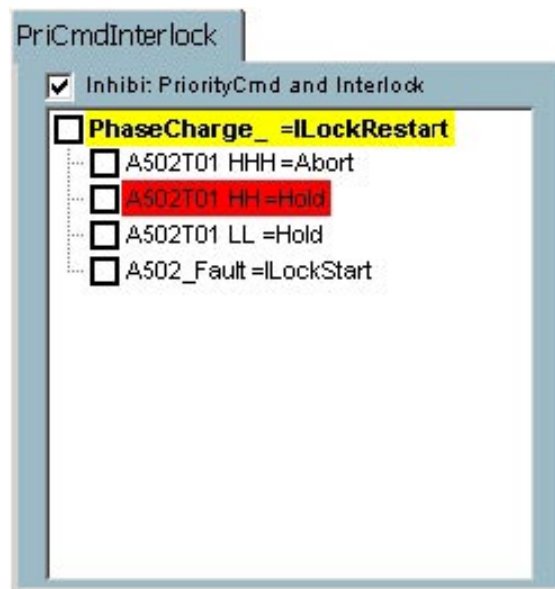


Figure 8. Operators know immediately why a Phase is in HOLD and cannot be resumed, simply by viewing the PCC Display on its faceplate.

The Industrial^{IT} Advantage

PCEquipmentLib is part of the Industrial IT[®] System 800xA suite of products, services and solutions from ABB. 800xA's Aspect Object technology and full integration with safety integrity systems, internal asset management, fieldbuses, intelligent motor control centres, drives and other automation and electrical devices provides many operational benefits while reducing cost and risk.

For the latest information on ABB visit us on the World Wide Web at <http://www.abb.com>



**For more information contact your local
ABB office or visit us at www.abb.com**

3BUS092091R0001 © Copyright 2005 ABB. All rights reserved.

Specifications subject to change without notice. Pictures, schematics, and other graphics contained herein are published for illustration purposes only and do not represent product configurations or functionality. User documentation accompanying the product is the exclusive source for functionality descriptions. The Industrial IT wordmark, Aspect Objects, and all above-mentioned names in the form XXXXXXIT are registered or pending trademarks of ABB. All rights to other trademarks reside with their respective owners.