Custom Steps

Build Manager supports the creation of "custom build steps" that can be used to add features to the product by including the step in any Promotion Path. Custom Steps allow organizations to address specific build needs by writing LotusScript code that can be invoked during the build at a specific point.

Because Custom Steps are run in the current context of the build, they run in client under the authority of the Notes ID (including Promote-as IDs) and have access to certain build artifacts, including the database currently being built, information stored in build variables (such as used in Macro replacements), and the Step documents that are part of the current build.

Creating a Custom Step consists of:

- Adding a new form to represent the UI for configuring the Custom Step
- Writing LotusScript to accomplish the desired action in a class that extends the class ActionExtension in the "Custom Action Extensions" script library
- Registering the new class in the function RegisterCustomActions in the "Custom Action Extensions" script library

Once a custom step is registered (Build Manager must be exited and reopened to pick up changes), Custom Steps can be added to a promotion from the left-hand navigator, by selecting the promotion path and choosing **Admin > Actions... > Custom...** and selecting the desired step.

Custom Steps can act on the following events:

- GetUserInput - called before the promotion switches to the background process to allow UI interaction if needed
- ValidateAction - called twice, once in the foreground process and once in the background process, so that the step can validate any pre-conditions
- ProcessCustomAction - called during the actual build (at the point where the step is included in the promotion) for the step to do it's primary work

Each event is passed appropriate parameters to provide easy access to build artifacts. All events provide access to the Step Document for the custom step, which can read settings, and be used as a starting point to locate other Step documents. ProcessCustomAction provides access to the database being created by the build.

Custom Step classes and methods are documented in the source code. Questions about creating Custom Steps should be directed to techsupport@teamstudio.com – note however that Teamstudio cannot support the custom code you write as part of an action, but only the framework itself.

Custom Action Utility class

Custom Step code can also make use of a utility class, ActionExtensionUtility that provides utility functions to simplify certain actions, such as logging, using build macros, and retrieving the Promotion Path and Database document. The class – along with documentation – is in the "Custom Action Extension" script library.

Using data collected by Custom Steps

In some circumstances it may be useful to retrieve data related to the current promotion and pass it on to the build log for the promotion, store it with a template created in Template Registry, or include it in an Approval request in the Build Manager Workflow database. One example of using this functionality might be to persist information about a Change Control request or approval made in an external system and related to the current promotion, and display it on the resulting build in the Template Registry.

As of Build Manager 8.1, the ActionExtensionUtility class includes the method SetWFProperty to allow String data to be stored in the build log and Template Registry documents, and passed to Approvals workflows.

The SetWFProperty function stores the String in a field named for the property name passed to the function, but prefixed with "WF" – for example, calling SetWFProperty with the name "ChangeControlNum" and the value "1234" will result in a text field named "WFChangeControlNum" on both the build log and on the Template document in Template Registry (if applicable), with the value "1234."

Both the Build Manager Report form (the promotion log) and the Database Template Information form in Template Registry contain a subform that can be customized to display this information and protected from design refresh to preserve it during upgrades. One possible use is pass a Notes URL via SetWFProperty and use code in the subform to open it via @URLOpen().

Build Manager Approvals workflows (based on Workflow.ntf provided with the Build Manager download) can also make use of these fields, by creating additional variables in the main Workflow Definition document and using a formula to set the workflow variable to the value of the WF-named field. Note that the workflow variable name should match the WF-named field to allow the eventual promotion to retrieve the value for display in the build log. For more information on customizing the Approvals workflow, contact techsupport@teamstudio.com.