ProjectBuilder

Table of contents

Generated by ControlTier
http://open.controltier.com/

1. Overview

ProjectBuilder is a utility type to assist developing and loading of source-based projects in the ControlTier system.

1.1. Formats for load-objects

1.1.1. Project XML format

Project XML documents are used both to specify the bulk loading of objects as well as define the format of dumps of sections of a Workbench project's object model.

The project XML document type (DTD) defines the most general XML mapping of the object model. The *project.dtd* and *sample.xml* files can be found in the "*lib/load-objects/projectxml*" directory of ProjectBuilder

Any number of objects can be defined in a given document. Each object is represented by one of the four base type elements: node, setting package and deployment. Each object of a given base type is identified by the required type and name attributes.

Note that unique object names by type are assumed, and that each file may reference a particular object only once. This means that the project XML format is not usable with modules that allow non-unique object names

When defining an object, it is unnecessary to specify all attributes beyond name and type. The system will preserve existing attribute values, or leave unspecified attributes of a new object unset. It is therefore possible limit the XML definition of an object to the type, name and only those optional attributes whose values must be changed.

The dependencies between objects are represented by the "resources" and "referrers" sub-elements. The "replace" attribute determines whether pre-existing resource and referrer relationships are preserved with the new being added, or whether they are wholly replaced by the new dependencies.

If the replace attribute is omitted, the resource and referrer dependencies are not replaced by those specified in the XML (i.e. the new dependencies supplement the existing configuration).

Note that batch operations associated with resources of all objects specified by the project XML are executed before those associated with referrers. This has particular relavence when replacing referrers since great care must be taken to ensure that a comprehensive set of replacements are provided. It is not sufficient to rely on the same dependency being specified

Generated by ControlTier
http://open.controltier.com/

as a resource relationship elsewhere in the project XML (or even more subtly to imagine that not replacing resource relationships for the parent object will automatically preserve the relationship from the referrer perspective).

Just remember that resources and referrers are two halves of the same relationship tieing the object model together as a whole!

However, if no resource or referrer dependencies are specified in the XML, then existing relationships are left undisturbed.

The "resource" sub-element identifies particular objects by type and name. These objects do not necessarily have to be defined in the XML document given that they already exist in the object model.

In general, semantic validation is left to the Workbench type model's constraint checking. Note that since new objects are created during processing of the document while updates are batched for execution once document processing is complete, a validation error at any point will leave the object model partially updated. i.e. there is no transactional integrity across the processing of a single document that contains new objects.

The net effect of this will be to leave extraneous new objects in the model.

However, documents that consist only of pre-existing objects will only use the batch update interface. Failures in this case are completely rolled back leaving the object model unchanged.

1.1.2. Tabular format

The tabular format is a delimited data format that has columns separated by the `-delimiter` argument value and records separated by newlines. Any character string can be used as a delimiter but the `:` (colon) character is default. Currently, there are two kinds of 'table' files used to declare instances of Deployment and Setting subtypes. A third can be used to declare object to object dependencies.

An example set of .tab files in a module directory structure are shown below:

- TypeX/
  - type.xml
  - commands/
  - objects/
  - - binding.tab
    - deployment.tab
    - setting.tab

Generated by ControlTier
http://open.controltier.com/

- templates/

Each format uses a different set of columns described as follows:

The deployment tabular format is used to define instances of Deployment subtypes.	

Format	`# deployment.tab format -`
	`#`
	`type:name:install-root:basedir:rank:description`
	`Apache:staging:/usr/local/apache2:/var/apache:3:t`
	`staging instance`

Example command usage	`ctl -p` *`project`* `-t ProjectBuilder -o`
	`object` `-c load-objects -- \`
	`-type TypeX -filename`
	`deployment.tab  -basetype deployment`

Table 1: Deployments

The setting tabular format is used to define instances of Setting subtypes.	

Format	`# setting.tab format -`
	`#`
	`type:name:settingValue:settingType:description`
	`ApacheDocroot:commonDocroot:/docroot:documentRoot`
	`docroot`

Example command usage	`ctl -p` *`project`* `-t ProjectBuilder -o`
	`object` `-c load-objects -- \`
	`-type TypeX -filename`
	`binding.tab  -basetype setting`

Table 2: Settings

The binding tabular format is used to define object child dependencies.	

Format	`# binding.tab format -`
	`#`
	`parentType,parentName:childType,childName`
	`# Apache ->`
	`ApacheSetting`
	`Apache,staging:ApacheDocroot,commonDocroot`

Example command usage	`ctl -p` *`project`* `-t ProjectBuilder -o`
	`object` `-c load-objects -- \`
	`-type TypeX -filename`

Generated by ControlTier
http://open.controltier.com/

	`binding.tab  -basetype binding`

Table 3: Bindings

2. About Commands

Command Overview

Given the number of commands available with ProjectBuilder, it may be difficult to get an overall view of what each command can do. The following table groups the commands into several categories and provides a link to the command's documentation.

Development Utilties	create-type, build-type, refactor-rename, convert-rdf
Data Utilties	generate-objects, load-objects, find-objects, purge-objects
Builder commands	build-library, generate-forrest-docs, load-types, runBuildScript
Jobcenter commands	load-jobs,find-jobs

3. See Also

Developer Reference for ProjectBuilder

Generated by ControlTier
http://open.controltier.com/