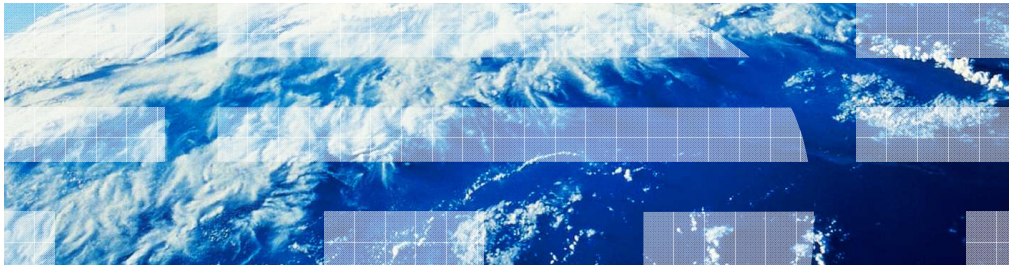


# IBM WebSphere Commerce Version 7 Feature Pack 2

## Content version handling



© 2011 IBM Corporation

This presentation will cover the content version handling feature that is included in the IBM WebSphere Commerce Version 7 Feature Pack 2.

## Agenda

- Overview of content version handling
- Workspaces
- Extending version handling
- Troubleshooting

The presentation will cover first an overview of content version handling and the content version handling features, content version handling within workspaces and then troubleshooting version handling.

## ***Overview of content version handling***

The first few slides will cover an overview of content version handling and the content version handling features.

## Version handling

- Manage versions of catalog data so that you can keep historic versions of catalog data for different promotional events
- UI enhancements in Management Center to create, restore and delete versions of catalog objects
- UI enhancements in Management Center to view previous versions of catalog objects



Content version handling is a new concept in Management Center. It allows you to capture a point in time version of an object and return to that version at a later date no matter how many times the object has changed since the version was made. This gives you the ability to rollback changes that are done in error, incorrect, or no longer needed. You can also make temporary changes to business objects to target specific activity events such as seasonal promotions. In Feature Pack 2, this version handling capability is available for catalog objects including categories, products, items, bundles, kits and sales catalogs.

The Management Center Catalogs tool allows you to create, restore, view and delete versions of your catalog objects. Versions also provide an audit history of catalog objects within WebSphere Commerce.

Version handling improves your productivity by removing the need to manually redo changes. It provides the capability to update catalog objects targeted for a promotional event and rollback to previous versions to increase effectiveness.

## Enable version handling

- Run enablement script from `WC_installdir/bin` or `WCDE_installdir\bin`
- Windows
  - `config_ant.bat -buildfile WC_installdir/components/common/xml/enableFeature.xml -DinstanceName=instance_name -DfeatureName=content-version -DdbUserPassword=db_password -DdbaPassword=dba_password [-DversionSchemaName=custom_versionschema_name] [-DdebugMode=true]`
- Developer/DB2/Oracle
  - `enableFeature.bat -DfeatureName=content-version -DdbUserPassword=db_password -DdbaPassword=dba_password [-DversionSchemaName=custom_versionschema_name] [-DdebugMode=true]`

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center.doc/tasks/enableversioncontrol.htm>

In order to enable version handling you need to run the enablement script. To enable version handling view the information center.

## Versions – best practices/concepts

- Versions are just snapshots and read-only
- You are always working on the current object
- If a version is created then the current object is based on the new version
- Before you make changes to a catalog object create a baseline
- If you choose to restore to another version, create a version based on your latest current changes

Versions are just snapshots and read-only, you cannot edit a particular version. If a version is created the current object will display that it is based on that new version, but you are always working on the current object and not working on the version. It is suggested that before you make changes to a catalog object that you create a baseline version so that at anytime you can go back to the baseline version. If you choose to restore to another version you should create a version based on your current changes and current object.

## Content version handling features

- Can be enabled in authoring or staging environment
- Configure objects to be included in a version
- Create versions of Catalog Objects of Master Category, Product, SKU, Bundle, Kit, Sales Catalog, and Sales Category
  - create a version
  - view a list of available versions
  - delete a version
  - restore a version
  - view a version
  - copy data from a version
- Visually compare current object with one of its available versions
- Update the name and description of version
- Remove versions of objects so that the version repository is of manageable size
- Documentation that explains how to include extensions to catalog objects in the version so that you can have historic versions of the extended business information

Content version handling can be enabled in an authoring and staging environment. Version handling allows you to create versions of catalog objects including master categories, products, SKUs, bundles, kits, sales catalog, and sales category. With version handling you can create, delete, and restore versions. You can manage a version of an object making short term changes and revert back after a promotional period is over. If you want to know what version the current version of a catalog object is based on you can view the object properties or view the list of all available versions. Within this list you can update names and descriptions of the versions. The management center has been enhanced so you can view current object with one of its available versions side by side. This allows you to copy data from a version of a object so that you can use information from a past version. Database commands exist to allow you to remove versions of objects to keep your version repository a manageable size. Feature Pack 2 also includes documentation that explains how to include extensions to product, SKU, category or catalog objects in the version so that you can have historic versions of the extended business information.

## Create a version of a category

The screenshot illustrates the process of creating a new version of a category. It shows a menu with a new 'Create Version' option highlighted. A dialog box titled 'Create Versions of Selected Business Objects' is open, allowing the user to enter a version name and description. Below the dialog, a toolbar contains icons for creating, deleting, restoring, and comparing versions. A table titled 'Available versions' displays the current version (Version 1) with a checkmark in the 'Base Version' column, indicating it is the current version.

Base Version	Version Number	Version Name	Creation Time	Created By	Description
✓	1	Baseline Girl Dresses and Skirts	2010/11/18 10:53 AM	wcsadmin	Baseline Girl Dresses and Skirts

1 of 1 selected

8 Content version handling © 2011 IBM Corporation

The screen captures on this slide highlight some of the key version handling features. A menu option has been added to allow you to easily capture your updates as a version. To create a new version, you provide a version name and description of your changes so you can easily find the version you want in the future.

When you view the properties page for a catalog object, a new tab is displayed that lists all the versions of the object. With the version handling toolbar icons you can create, delete, restore versions, and easily compare versions with the current object. You can also edit the version name and description of the version.

You can tell what version the current version of a category is based on. You can view the list of versions of a category and look for the 'check mark' besides a version that indicate what the current version is based on. You can also look at the object properties of the catalog object.



## View a version

- Double-click version

Category Associated Assets **Versions** References

Available versions ⓘ

Base Version	Version Number	Version Name	Creation Time	Created By	Description
✓	1	Baseline Girl Dresses and Skirts	2010/11/18 10:53 AM	wcsadmin	Baseline Girl Dresses and Skirts

1 of 1 selected

- View version in read-only mode

Girls Dresses and Skirts(v1) (Read-Only)

Manage Category Associated Assets References

**General Category Information**

\*Code ⓘ Girls Dresses and Skirts

\*Name (United States English) Girls Dresses and Skirts



Short description (United States English) Dresses and skirts

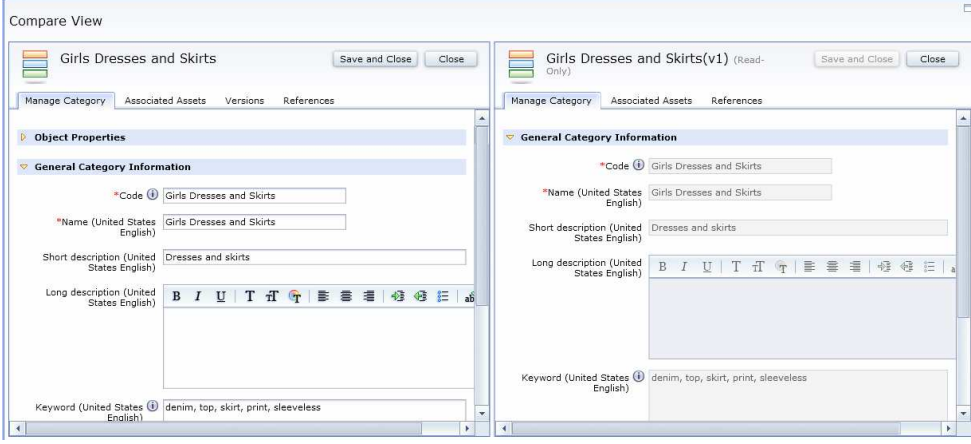
Long description (United States English)

B I U T f T [Icons]

To view a version of a catalog object so that you can check what was changed in last promotion. First select a catalog object and open the properties view of the object and select the Versions tab. Select a version and double-click the selected version or select the open menu item under Edit menu or in the pop-up menu to open the properties view of the selected version of the catalog.

## View current object with version side by side

- Split view icons 
- Click compare version with current object toolbar icon 



10 Content version handling © 2011 IBM Corporation

Another new feature of the Catalogs tool that is useful in working with versions is the side by side view. Your main work area in Management Center can be split either horizontally or vertically so two objects can be displayed at a time. While you can use this view mode for any two objects, it is particularly useful for comparing two versions of an object or comparing a version to the current object. The first snapshot shows the split view icons. For your convenience, within the version toolbar there is a compare version with current object icon. After selecting that you will see a side by side view of the current object and one of its versions. You can cut and paste text from the versioned object into the current object.

## Remove versions of objects

- `dbclean -object content_version -type object_type -dbpasswd pswd -instancexml file -days #`

object_type	Description	content_version
inactive	Delete the versions that are number of days old and inactive	
catalog	Delete all the old versions of sales catalog based on the create time. Keep the active version only	SalesCatalog
catalog_group	Delete all the old version of catalog group based on the create time. Keep the active version only	CatalogGroup or SalesCatalogGroup
catalog_entry	Delete all the old version of catalog entry based on the create time. Keep the active version only	Product, ProductSKU, CatalogGroupSKU, Bundle, Kit

- Example
  - `dbclean -object content_version -type inactive -dbpasswd wcs1 -instancexml C:\Websphere\CommerceServer70\instances\demo\xml\demo.xml -days 10`

11

Content version handling

© 2011 IBM Corporation

You can remove versions of objects so that your repository is of manageable size. You can cleanup the inactive versions by running dbClean on object content version and type inactive and specify in the command parameters the number of days old for these inactive versions for deletion. Inactive versions that met the number of days old requirement are deleted. Active versions are not deleted. You can similarly clean up expired versions of catalog, expired catalog group versions, and expired catalog entry versions.

The different commands have different choices for the content version. For Catalog, you need to clean the object type: SalesCatalog. For Catalog Group, you need to clean the object type: CatalogGroup or SalesCatalogGroup. For Catalog Entry, you need to clean the object type: Product, ProductSKU, CatalogGroupSKU, Bundle, or Kit.

The slide shows an example to clean all the inactive versions that are 10 days or older.

Section

## ***Workspaces***

The next section will cover version handling within Workspaces.

## Using content version handling in workspaces

- You can also create, restore, view, and delete versions of business objects within a workspace
- You can restore different versions on different workspaces
- A workspace or task group can work on different copies of the object targeted to be completed in a different time frame or for different purposes
- Versions are shared amongst all workspaces and approved content
  - For example: if you create a version of an object in a workspace, it can be immediately viewed in approved content
- It is suggested that you create a baseline version before you edit objects and create a version after you edit objects

Content version handling capabilities can also be extended for use in WebSphere Commerce Workspaces. You can create, restore, view, and delete versions of business objects within a workspace. With workspaces in place, an object can be active in multiple areas: the production ready schema and in a workspace. Objects can be versioned in either workspace or in the production ready state and be available for either to revert to. A workspace or task group is able to work on different copies of an object targeted to complete in a different timeframe or for a different purpose. Once completed the workspace version of the object will replace the active object in the production ready schema. You can restore different versions on different workspaces. So different workspaces can have different "based on version" of the same catalog object to work on. Versions are shared amongst all workspaces and approved content, for example: if you create a version of an object in a workspace, it can be immediately viewed in approved content.

One can foresee a business user restoring different versions and changes for different promotions on workspaces. So one workspace can have October changes and one for November changes. At anytime you can put different workspaces changes to the base content or push to production and have a way to restore back to the same baseline.

## Work on task to create a holiday promotion

To Do - Task List

* Type	* Code	Name	Status	Due Date	Task Group	Workspace
	T_10002	Create Holiday Promotion		2010/09/04	Create new promotion	Workspace 1

▪ Create a baseline version first, edit the current object, and create a new version after your edits

A0000372
Save and Close Close

Manage Product
Descriptive Attributes
Defining Attributes
Merchandising Associations
Associated Assets
Versions
References

Available versions

Base Version	Version Number	Version Name	Creation Time	Created By	Description
	1	Create Baseline Version of Rebuilt Master Cylinder	2010/09/03 03:40 PM	wcsadmin	Create Baseline Version of Rebuilt Master Cylinder
✓	2	Holiday Promotion Version	2010/09/03 03:46 PM	wcsadmin	Holiday Promotion Version

0 of 2 selected

14
Content version handling
© 2011 IBM Corporation

The next few slides will show how to change an object for a promotional update and then revert the object back to its baseline version, using workspaces. A workspace is created with two task groups each with a task. One task is to create a holiday promotion, the other is to revert the object back to the baseline version before the promotional material was added. You start off by working on the to do task of creating a holiday promotion. Go and create a baseline version first of the object you are editing. Edit the current object and then create a new version after you have done your editing. Approve the task and activate to production. Then view the object in the storefront and notice the updates.

## Revert holiday promotion back to baseline version

To Do - Task List

* Type	* Code	Name	Status	Due Date	Task Group	Workspace
	T_10006	Revert promotion back to baseline		2010/12/26	Revert back to baseline	Workspace 1

■ Restore baseline version

Base Version	Version Number	Version Name	Creation Time	Created By	Description
	1	Create Baseline Version of Rebuilt Master Cylinder	2010/09/03 03:40 PM	wcsadmin	Create Base Rebuilt Mast
<input checked="" type="checkbox"/>	2	Holiday Promotion Version	2010/09/03 03:46 PM	wcsadmin	Holiday Pron

1 of 2 selected

Are you sure you want to restore to Version 1 and overwrite the current version of this object?

A0000372

[Manage Product](#)
[Descriptive Attributes](#)
[Defining Attributes](#)
[Merchandising /](#)

**Object Properties**

Based on version 1

Version name Create Baseline Version of Rebuilt Master Cylinder

15
Content version handling
© 2011 IBM Corporation

Now that you made the promotional updates, go and work on the task to revert the object back to the baseline version. Restore the baseline version. Complete the task, get the task approved and activated to production. The object will now be back to the baseline version.

## ***Extending version handling***

The next slides will cover how to extend version handling.



## Content version handling configuration file 1 of 2

- The content version handling configuration file, `wc-content-version.xml`, defines the configuration for the UI objects that are versionable

– Code snippet:

```
<wc:ContentVersionNoun name="CatalogEntry" topTable="CATENTRY"
  className="com.ibm.commerce.catalog.version.CatalogEntryContentVersionServiceImpl">
<wc:ConfigGroup id="-1">
<wc:ContentVersionUIObject name="Product" />
<wc:ContentVersionRelatedTable name="CATENTRY" whereClause="CATENTRY_ID=? AND MEMBER_ID=?"
  values="{objectId}, {storeOwnerId}"/>
<wc:ContentVersionRelatedTable name="CATENTDESC" whereClause="CATENTRY_ID=?"
  values="{objectId}"/> ...
```

- If a service module manages UI objects that are versionable, it should have a `wc-content-version.xml` configuration file

– The configuration file is located in `WC_eardir\xml\config\servicemodulename\` or `\servicemodulename-fep;`

- Catalog service module configuration file is located `WC_eardir\xml\config\com.ibm.commerce.catalog\` directory or `\com.ibm.commerce.catalog-fep\` directory

– You must not modify the `wc-content-version.xml` file in either of the preceding directories

The content version handling configuration file defines the configuration for the UI objects that are versionable. This file tells you which UI objects are versionable, and how the system versions the UI object. A snippet of the `wc-content-version.xml` file is shown here. This code if continued will describe the version handling for Product, ProductSKU, CatalogGroupSKU, Bundle, and Kit UI objects as they all map to the CatalogEntry noun.

If a service module manages UI objects that are versionable, it should have a `wc-content-version.xml` configuration file. In this file, you specify which parts of the UI objects to save when creating a version. The path for the configuration file is shown here where `servicemodulename` is the name of the service module. An example of the Catalog service module configuration file is shown here too. You must not modify the `wc-content-version.xml` that comes with the feature pack because WebSphere Commerce fix pack updates might overwrite your changes. Visit information center for more information.

## Content version handling configuration file 2 of 2

- Customizing content version handling
  - To change the default configuration for versionable objects
    - Create an extended configuration file `wc-content-version.xml` and place it `WC_eardii\xml\config\com.ibm.commerce.servicemodule-ext\`
      - `com.ibm.commerce.servicemodule-ext` contains all the extended files for a particular service module
- Extend the content version handling feature
  - Create a content version handling configuration file in your service module directory for each custom object you want versionable
  - Ensure the custom tables you want to version exist in the base schema.
  - Create `wc-content-version.xml` in `WC_eardii\xml\config\com.mycompany.commerce.servicemodule\`
  - Build your custom configuration file

The content version handling configuration file defines the configuration for the UI objects that are versionable. To change the default configuration for versionable objects, you must extend the configuration file by creating a file in the ext folder. The extended configuration file `wc-content-version.xml`, must be placed in the directory shown here. Where `com.ibm.commerce.servicemodule-ext` contains all the extended files for a particular service module. Note: This file must contain only the custom changes.

You can extend the content version handling feature and create a content version handling configuration file in your service module directory for each custom object you want versionable. This file defines the configuration for the UI objects that are versionable and what parts of the UI objects to save when a version is created. First, ensure the custom tables you want to version exist in the base schema. Next create a `wc-content-version.xml` in the path shown here. Note: You must name your content version handling configuration file `wc-content-version.xml`. Then build your custom configuration file. See the information center for more details.

## The updateVersionSchema utility

- The updateVersionSchema utility creates, in the version schema, corresponding tables that exist in the base schema and that are marked to be versioned in wc-content-version.xml
- Run this utility
  - If you customize any of the tables listed to be versioned by the content version handling feature or
  - When you make any schema-related updates to the list of versionable tables in the base schema
- Ensure that you have an updated content version handling configuration file that includes the custom tables
- Stop the WebSphere Commerce test server
- Run updateVersionSchema

The content version feature enablement will automatically update workspace schemas if the content management feature has been enabled. For any additional base schema change, workspace schemas for the content management feature must be updated separately as well in addition to running the UpdateVersionSchema utility. You will run this utility if you customize any of the tables listed to the versioned by the content version handling feature or when you make any schema-related updates to the list of versionable tables in the base schema. The updateVersionSchema utility creates, in the version schema, corresponding tables that exist in the base schema and that are marked to be versioned in wc-content-version.xml. Only the primary key, and unique indexes from the original table in the base schema are replicated in the table created in the version schema.

To determine which tables to create in the version schema, the updateVersionSchema utility searches all the service module directories for the content version handling configuration file (wc-content-version.xml). For any tables listed in the wc-content-version.xml file, the updateVersionSchema utility looks up the existing table in the WebSphere Commerce base schema, and then creates a corresponding table in the version schema. If the table listed in the wc-content-version.xml file is not found, an exception is reported.

## User customization documentation

- Sample code, tutorial and documentation to support user customization
- In the Warranty tutorial, you can customize the Catalog service to support warranty information and care instructions for Catalog Entries

[http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center\\_customization.doc/tutorial/tfadpcat.htm](http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center_customization.doc/tutorial/tfadpcat.htm)

- To continue with that tutorial, documentation has been created to demonstrate how to version an existing object CatalogEntry to include user data Warranty

[http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center\\_customization.doc/tasks/tfcontentversioncustomdata.htm](http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center_customization.doc/tasks/tfcontentversioncustomdata.htm)

- To extend the content version handling feature to support UserData in your existing noun
  - Identify which service module and UI objects you want to customize
  - Update the content version handling configuration
  - Updating the version schema with the custom tables
  - Viewing userData versions on the UI

You can extend the content version handling feature to support UserData in your existing noun. Currently, content version handling is enabled only for the Catalog service module.

The information center includes documentation that explains how to include extensions to product, SKU, category or catalog objects in the version so that you can have historic versions of the extended business information. Previously there is a Warranty tutorial, with this you can customize the Catalog service to support warranty information. Care instructions for Catalog Entries. Warranty information is added to the CatalogEntry noun as a UserData element to demonstrate the addition of language independent property while care instruction is added to CatalogEntryDescription noun part as an attribute element to demonstrate the addition of language dependent property.

To customize existing nouns, first you will identify which service module and UI objects you want to customize. Then you will need to perform one or more of these steps. Updating the content version configuration file. Next update the version schema with the custom tables using updateVersionSchema. Finally you will want to be able to view the userData on the UI, by customizing both the client side and server side. After you complete this customization, whenever you create a new version against a product your newly added user data Warranty is versioned as well.

## Extending the content version handling feature to support additional business objects

- To customize your custom objects or existing WebSphere Commerce objects, perform these actions:
  - Create a content version handling configuration file
  - Update the version schema with custom tables
  - Implement access control for version services
  - Create a new tab to display version information
  - Enable the version services for custom object
  - Update the JSP and query template to enable viewing custom object versions on the UI

You can extend the content version handling feature to support version handling in your own custom objects or in existing WebSphere Commerce objects. To customize your custom objects or existing WebSphere Commerce objects, perform these actions. First create a content version handling configuration file. You create a content version handling configuration file in your service module directory for each custom object you want versionable. This file defines the configuration for the UI objects that are versionable and what parts of the UI objects to save when a version is created. Next update the version schema with custom tables using the `updateVersionSchema` utility. Then implement access control for version services. Next create a new tab to display version information. You add a new tab to show the list of versions for your custom object. To define the new tab named "Version", you update your business object properties definition file. Then enable the version services for custom objects. You update the primary object definition for the custom object you want to version. In the primary object definition file, you add the functionality to create versions, update versions, and retrieve the list of versions for your custom object.

Update the JSP and query template to enable viewing custom object versions on the UI to enable viewing different versions on the UI, you customize both the client side and server side. For more detail information visit the information center.

Section

## ***Troubleshooting***

The next section will cover troubleshooting of version handling.

## Trace strings

- The key trace components for debugging content-version are:  
`com.ibm.commerce.content.*=all: com.ibm.commerce.context.version.*=all:`
- For any web services related problem:  
`com.ibm.commerce.foundation.*=all`
- For any workspace related problem:  
`com.ibm.websphere.commerce.WC_CONTENT=all`

The key trace components for debugging content-version are shown here. Traces for web services and any workspace related problems are also shown here.

## Content-version feature enablement

- During content-version feature enablement, log files and SQL log files are located under
  - <CommerceServer70>/logs
  - Runtime: <WAS\_installdir>/instances/<instance\_name>/logs
  - Toolkit: <WCDE\_installdir>/logs directory
- In case of feature enablement failure, turn on debug option to re-run the feature enablement task
  - The following are all the possible log files generated:
    - Create\_version\_db\_<instanceName>.sql.log
    - enablecontent-version\_<timestamp>.log
    - enableFeature.log
    - Application.version.enablement.messages\_<timestamp>.log

During content-version feature enablement, log files and SQL log files locations are shown here.

In case of feature enablement failure, turn on debug option to re-run the feature enablement task. The log files displayed here are all the possible log files generated.



## Content-version feature enablement – updateVersionSchema

- UpdateVersionSchema utility to update the version schema must be used for
  - Any additional schema update for content-version customization after content-version feature enablement
  - Updating the corresponding tables in the workspace schemas
  - Enabling more tables in the wc-content-version.xml
- Potential log files generated by the updateVersionSchema utility are:
  - UpdateVersionSchema.log
  - Update\_version\_db\_<instanceName>.sql.log
  - Application.version.update.messages\_<timestamp>.log

The feature enablement step will update workspace schemas once if the content-management feature has been enabled before any feature enablement. After content-version feature enablement, any additional schema updates for content-version customization will mandate the use of UpdateVersionSchema utility to update the version schema. In addition, the UpdateWorkspacesSchema utility must be used to update the corresponding tables in the workspace schemas. When you enable more tables in the wc-content-version.xml, you also need to run the UpdateVersionSchema to add these new tables in the version schema.

Version schema cannot be created twice. If there is any schema with the same name as the target version schema name, it must be completely removed from the database before enabling the content-version feature. Potential log files generated by the UpdateVersionSchema utility are listed here and require the debug option to be set.

## View version object - suppress unversioned contents from catalog entries

- Suppress the unversioned contents
- Example: For Merchandising association tab, find the property definition of 'Merchandising Association' tab of the product ProductPropertiesView.def
  - Add a new EnablementCondition
  - The completed PropertyTabPane is:

```
<PropertyTabPane name="productMerchandisingAssociationsTab"
  text="{catalogResources.productMerchandisingAssociationsTab}">
  <EnablementCondition enablementValue=""
    propertyName="objectId"
    conditionId="noObjectIdCondition"/>
  <!-- Property Pane: Merchandising Associations. This is an
  instantiation of the property pane class which contains
  merchandising associations details. -->
  <PropertyPane
    baseDefinitionName="catProductMerchandisingAssociations" />
</PropertyTabPane>
```

You might want to suppress unversioned content from catalog entries when you view a version object. If you have a primary object, for example, a product. While viewing a version of this product, you might not want users to see the unversioned contents. To suppress the unversioned contents, for example Merchandising association tab, you can do the follow these steps. First find the property definition of 'Merchandising Association' tab of the product in ProductPropertiesView.def. Then add a new EnablementCondition, so that the tab is hidden/suppressed while viewing a product version. The completed PropertyTabPane is shown here. This enablement condition is evaluated to "true" only when the containing primary object is not a version object. Later on, if you add the support to version the content, you can remove this enablement condition, so that the Tab will show up again.

## Customization – view versions 1 of 2

- Problem: Not all data can be viewed successfully when you customize "view versions"
- Example: When you view the different recipe data, recipe ingredient is not correct, but recipe instruction is correct in the corresponding version
- Check if you are getting the right query, defined in the tpl file
- Determine which SQL you need do the these steps:
  - Log into CMC with Firefox and Firebug enabled, view a version on the Recipe's Version tab, click the ingredient tab and grab the request link from Firebug
    - Example request link: `https://cvt06:8000/lobtools/cmc/GetRecipeChildren-Ingredients?requestIdentifier=2&timeZoneId=America%2FNew_York&objectVersionId=10010&locale=en_US&identityId=15146&storeId=10202&maxItems=25&dataLanguageIds=-1&parentId=2004&recipeId=2004`
  - Search struts file by using GetRecipeChildren-Ingredients and locate the JSP
 

```
<action path="/GetRecipeChildren-Ingredients"
  forward="/jsp/mycompany/recipe/GetRecipeChildren-Ingredients.jsp" />
```
  - In the JSP, make sure the VersionID parameter is added in, and find out the expressionBuilder information: `expressionBuilder="getProjectMaterial"`

When you customize your "view versions" , you can have the issue that you cannot get the data to be viewed successfully. To walk through this, lets look at viewing different recipe data. The recipe tutorial is located in the information center, and existed before the feature pack. The example issue is that the recipe ingredient is not correct with the corresponding version, while the recipe instruction is correct. In this case, the issue is that you are not getting the right query versioned which is defined in the tpl file.

In order to determine which SQL you need to make it be versioned for viewing, you need to do these steps. First log into the Management Center by using Firefox, and navigate to the Recipe's Version tab, and choose one version to be viewed. You will notice that the recipe ingredient's data is not versioned properly when you click the ingredient tab to get the related information. If you enable Firebug, then you can get the request link. Next search the struts file by using GetRecipeChildren-Ingredients (notice this was in the request link) to locate the corresponding JSP. In the JSP GetRecipeChildren-Ingredients.jsp, make sure the VersionID parameter is added in, and find out the expressionBuilder information `expressionBuilder="getProjectMaterial"`.

## Customization – view versions 2 of 2

- Go to get-data-config.xml which is under WEB-INF\config\com.mycompany.commerce.project and find out the corresponding access profile:

```
<name>accessProfile</name>
<value>MyCompany_Admin_ProjectMaterial</value>
```
- Go to query tpl file wc-query-Project-get.tpl and search MyCompany\_Admin\_ProjectMaterial to get the associated query:

```
associated_sql_statement=MyCompany_ProjectMaterial
```
- Search the query name to get the associated query:

```
BEGIN_ASSOCIATION_SQL_STATEMENT
  name=MyCompany_ProjectMaterial
  param=versionable
  base_table=XPROJECT
  sql=
```

  - Make sure param=versionable is added in, then restart the server
- You should now see the right recipe ingredient data

Continue on and go to Get-data-config.xml. There you can find out the corresponding access profile with the expressionBuilder information you got from the previous step. Next go to query tpl file and search MyCompany\_Admin\_ProjectMaterial to get the associated query. Be sure param=versionable is added in. After you do that, restart the server, then you can see the right recipe ingredient data.

## Objects written into database correctly?

- On the Developer machine, open a browser to URL:

<http://localhost/webapp/wcs/admin/servlet/db.jsp>

- Enter this SQL and click the "Submit Query" button:

```
select * from cmversninfo;
```

- Optionally you can refine the SQL with a WHERE clause, for example:

```
select * from cmversninfo where name='my recipe version';
```

- Note the `cmversninfo_id` returned from the previous query and use it to look at the versioned data, for example:

```
select * from vcw_demo.xproject where cmversninfo_id=?;
```

```
select * from vcw_demo.xprjmtr where cmversninfo_id=?;
```

```
select * from vcw_demo.xprjins where cmversninfo_id=?;
```

If you want to check if the objects data is written into the database correctly, you need to first on the server machine, open a browser to the URL shown here. Then enter `select * from cmversninfo;` for SQL and click the "Submit Query" button. Optionally you can refine the SQL with a WHERE clause, like the examples shown above. You can use the `cmversninfo_id` returned from this query and use it to look at the versioned data.

## Errors with wc-content-version.xml

- During customization, you need create your own wc-content-version.xml
  - It is easy to make errors, for example in
 

```
<wc:ContentVersionRelatedTable name="XPRJCOL"
  whereClause="XPRJCOL_ID=? AND STOREENT_ID=?"
  values="{objectId}, {storeId}/>
```

    - You missed }" here
  - View this file in Internet Explorer to verify there are no errors
- When you log into CMC you will get the error msg like: An error has occurred while attempting to communicate with the server
  - If you looking into the log in the console panel, you will notice:
 

```
com.ibm.commerce.foundation.internal.server.command.impl.DefaultCommandFact
orySourceImpl createCommand() CWXEF0029E: Failed to find the command
implementation for interface
com.mycompany.commerce.project.facade.server.commands.FetchProjectCollectio
nCmd and key /ProjectCollection
[11/11/10 14:04:20:155 EST] 00000010 config          E
com.ibm.commerce.content.config.ContentVersionConfigHelper validateXml
org.xml.sax.SAXParseException: The value of attribute "values" associated
with an element type "wc:ContentVersionRelatedTable" must not contain the
'<' character.
```
- You will know the error deals with something with "ProjectCollection", be sure to check the syntax in wc-content-version.xml

During customization, you need to create your own wc-content-version.xml, it is easy to make some typing error or usage-error. There is an example shown here where a }" is missing. You can view this file in Internet Explorer to verify there are no syntax errors. If however, you miss the error, when you log into CMC, you will get an error message like: An error has occurred while attempting to communicate with the server. You will notice that there might be a server error. If you look into the log, in the console panel, you will notice the error shown here. With this error, you will know something is wrong with "ProjectCollection", then you can further check the corresponding syntax in the wc-content-version.xml.

## Summary

- Manage versions of catalog data so that you can keep historic versions of catalog data for different promotional events
- You can create, restore, view, and delete versions of business objects
- You can view a version and current object side by side
- With versions you can:
  - Roll back changes
  - Make temporary changes
  - Use for an audit trail

With the version handling feature included in the WebSphere Commerce V7 Feature Pack 2 you now can manage versions of catalog objects so that you can keep historic versions of catalog objects for different promotional events. You can create, restore, view, and delete versions of categories, products, SKUs, bundles, kits, and sales catalog. You can view a version and current object side by side, and copy data from a version to the current object. Version handling allows you to roll back changes, make temporary changes. You can use version handling for an audit trail.

## References

- **Enable version handling**

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center.doc/tasks/tfenableversioncontrol.htm>

- **Working with content version handling**

[http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center.doc/tasks/tpr\\_cvmain.htm](http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center.doc/tasks/tpr_cvmain.htm)

- **Customizing content version handling**

[http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center\\_customization.doc/tasks/tfcustcontentversion.htm](http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center_customization.doc/tasks/tfcustcontentversion.htm)

- **Warranty tutorial**

[http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center\\_customization.doc/tutorial/tfadpcat.htm](http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center_customization.doc/tutorial/tfadpcat.htm)

- **Tutorial: Extending the Recipe tool to support content version handling**

[http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center\\_customization.doc/tutorial/tfversionrecipe\\_intro.htm](http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center_customization.doc/tutorial/tfversionrecipe_intro.htm)

Here is a list of references.





## Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send email feedback:

[mailto:iea@us.ibm.com?subject=Feedback\\_about\\_Versioning.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_Versioning.ppt)

This module is also available in PDF format at: [../Versioning.pdf](#)

You can help improve the quality of IBM Education Assistant content by providing feedback.



## Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "[Copyright and trademark information](http://www.ibm.com/legal/copytrade.shtml)" at <http://www.ibm.com/legal/copytrade.shtml>

Windows, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

JSP, and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.

© Copyright International Business Machines Corporation 2011. All rights reserved.