

WebSphere Commerce V7.0

Development and test tools



This presentation introduces updates to development and test tools in WebSphere® Commerce version 7.

Goals

- To understand how WebSphere Commerce Developer has changed in V7
- To understand the new Management Center test automation framework and how to use it

At the end of this presentation, you should understand the changes made to WebSphere Commerce Developer for V7. You should also understand how to use the new Management Center test automation framework.

Agenda

- Development environment updates
- Web services development
- Management Center testing

First, an overview of development environment updates is provided, followed by a detailed discussion of changes to Web services development. The second half of the presentation focuses on Management Center testing.

Development environment updates

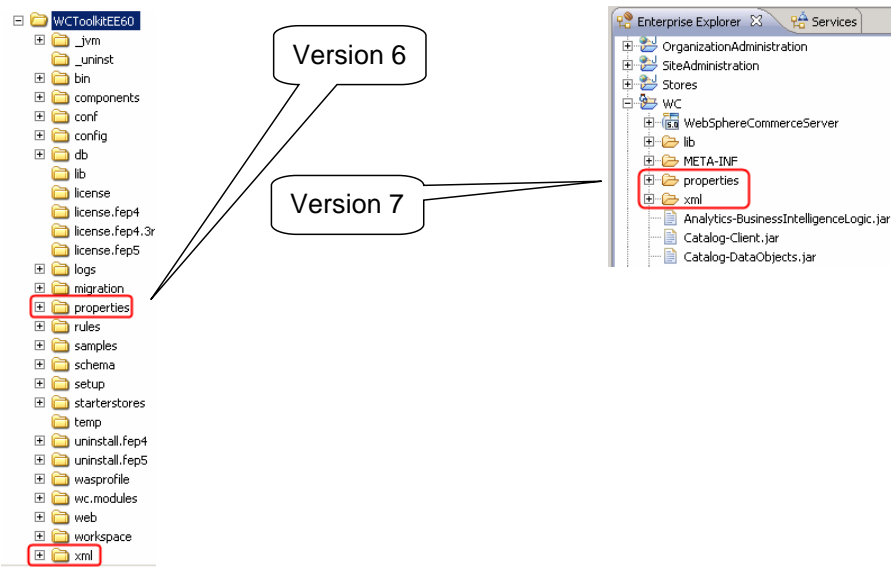
This section reviews development environment updates.

Overview

- More files contained in workspace
- Cloudscape renamed to Derby
 - Cloudscape references still exist
- JET used for Web service development

The capabilities of WebSphere Commerce Developer have not changed significantly in version 7. Two changes to note are that properties and XML configuration files are now contained within the workspace. Also, the development database Cloudscape has changed name to Derby and is now an Apache open-source project. Some references to Cloudscape still exist in the environment. There is no change to the actual development database; just the brand name has changed.

More files contained in workspace



In version 6, the properties and xml folders existed outside of the Rational® Application Developer workspace. This meant switching in and out of Rational Application Developer to perform many configuration and customization tasks. In version 7, the properties and xml folders are contained within the workspace in the 'WC' application project.

Changes in Web service development

This section introduces changes in Web service development.

Web services development

- Version 6
 - Download Design Pattern Toolkit (DPTK) from AlphaWorks and update Rational Application Developer
 - Download ComponentProjects file from Information Center and load into workspace
- Version 7
 - Install Java™ Emitter Template (JET) with Rational Application Developer

In WebSphere Commerce version 6, the design pattern toolkit (DPTK) is used to simplify Web services development. Getting set up to start Web services development requires extra steps, such as downloading the toolkit from AlphaWorks. Associated WebSphere Commerce assets also have to be downloaded from the Information Center and loaded into Rational Application Developer. In version 7, getting started with Web services development is made easier with the adoption of the Java Emitter Template (JET). JET is a built-in feature of Rational Application Developer 7.5 and can be installed as part of your Rational Application Developer installation process.

JET installation

- Installed as part of Rational Application Developer 7.5



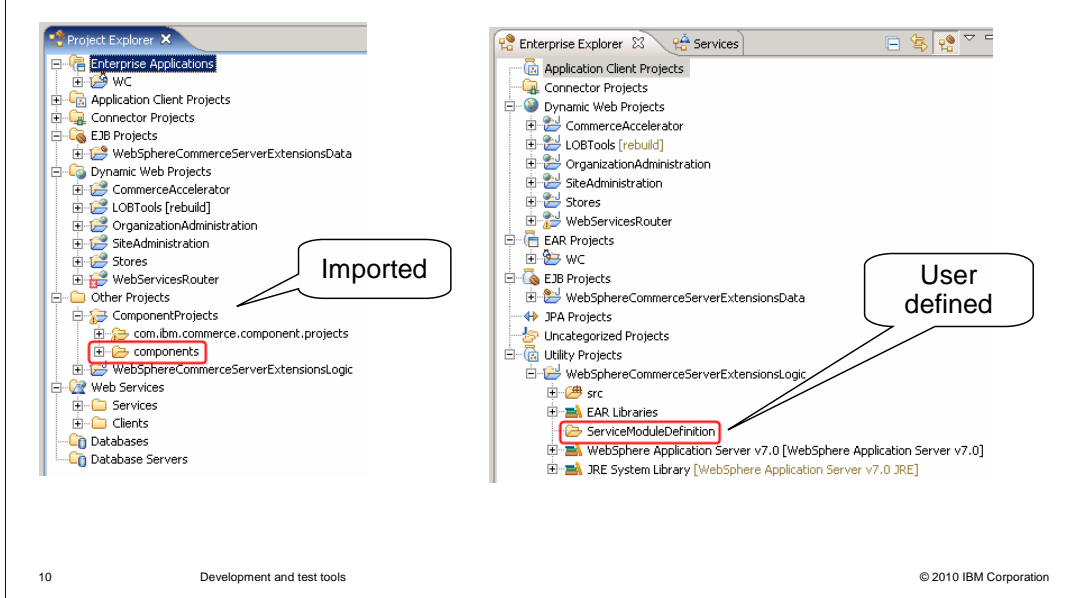
9

Development and test tools

© 2010 IBM Corporation

The JET Transformation Authoring and Runtime component needs to be selected during the Rational Application Developer 7.5 installation. If this option is overlooked, you can choose to install the package at a later time. The Information Center has details on installing JET into an existing WebSphere Commerce Developer environment.

Application definition location



WebSphere Commerce uses the JET plug-in for creating WebSphere Commerce service modules from a simple XML file as it did with DPTK. By describing the service module in a specialized XML syntax, the service modules can be generated. This allows you to start directly with the service module implementation without having to spend hours with the setup and configuration of a service module. The location of these XML files, known as application definitions, has changed slightly. In version 6, the application description files are stored within the ComponentProjects project that is imported after installing DPTK. In version 7, your application definition files should be stored in the WebSphereCommerceServerExtensionsLogic project. Create a directory to store the files. In this screen capture, the new directory is called ServiceModuleDefinition.

The format of the application definition file has changed only slightly. The Web services tutorials in the Information Center have all been updated to show the steps for creating services using JET.

Application definition

Version 6

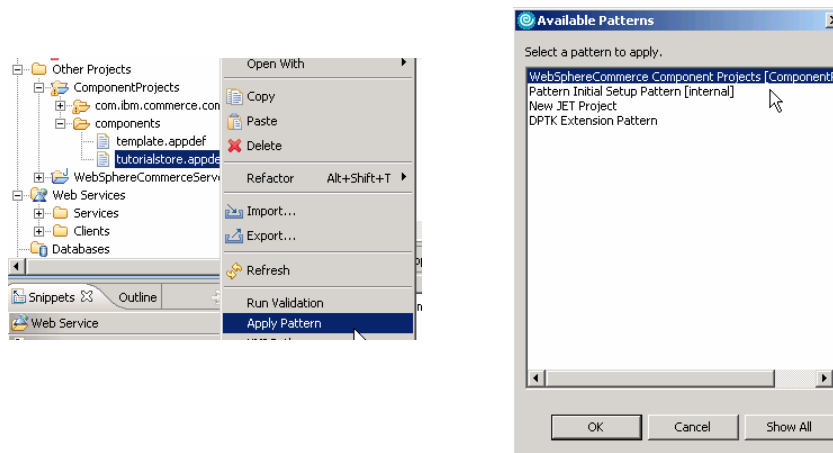
```
<commerce-component name="TutorialStore" company="MyCompany"
  packagenameprefix="com.mycompany.commerce"
  namespace="http://www.mycompany.com/xmlns/prod/commerce/9"
  nlsprefix="mycompany" type="SOI">
  <noun name="TutorialStore" get="true" process="true" change="false" sync="false"/>
</commerce-component>
```

```
<_pattern:commerceComponent xmlns:_pattern="http://www.ibm.com/xmlns/prod/commerce/foundation/pattern"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.ibm.com/xmlns/prod/commerce/foundation/pattern
  ../../WC/xml/config/xsd/wc-component-patten.xsd "
  name="SOITutorialStore" packagenameprefix="com.mycompany.commerce" company="MyCompany"
  namespace="http://www.mycompany.com/xmlns/prod/commerce/9/soitutorialstore" nlsprefix="myco" type="SOI">
  <_pattern:noun name="TutorialStore" get="true" process="false" change="false" sync="false"/>
</_pattern:commerceComponent>
```

Version 7

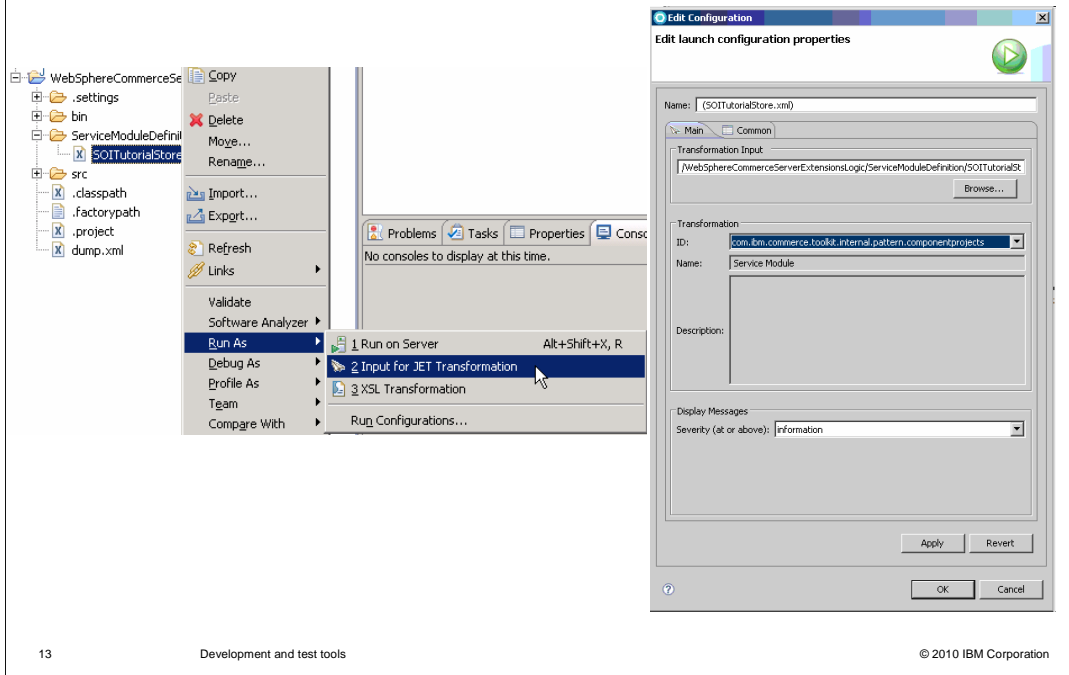
An application definition is a simple XML file that is fed into a modeling tool. The definitions used by JET look very similar to those used by DPTK. The most significant difference is the use of XML namespaces.

Generating service module with DPTK



Here you can see the steps in version 6 to generate a service module using DPTK.

Generating service module with JET



13

Development and test tools

© 2010 IBM Corporation

Note the menu options and UI have changed for version 7. To perform a transformation using JET, you need to select “Input for JET Transformation” from the “Run As...” sub menu.

Management Center test automation framework

This section introduces the Management Center test automation framework.

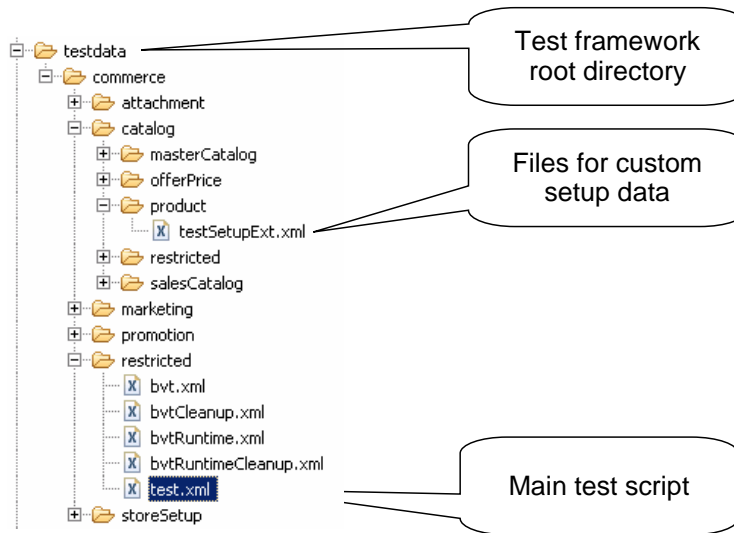
Framework overview

- XML based test cases
- Support for test cases that validate
 - Management Center model
 - Services
 - Object definitions
- Does not support UI testing

`https://<host>:8000/lobtools?testdata=/testdata/commerce/mycompany/test.xml`

The Management Center test automation framework allows you to write XML-based test cases that validate the Management Center model, services, and object definitions. The framework does not address validation of Management Center UI widgets and requires a manual login to Management Center to initiate testing. The test automation framework uses an XML file that describes the list of actions you want to run. You pass the XML file to Management Center by adding it as a URL parameter when launching the tool. A sample URL is shown at the bottom of this slide.

Directory structure



The testdata folder contains all of the pre-defined test scripts. It can be found under in the LOBTools project under the WebContent directory. Each tool has its own directory and test files. External setup files are available for the different business objects. You can add custom initialization data to these files and it will be included in the predefined tests. The main test file is located in the restricted directory. This file launches the test scripts for each of the Management Center tools. You can also create your own test scripts and store them in a custom directory under the testdata directory.

Test case structure

```
<testcase>

  <!-- Run test against B2B store -->

  <!-- Set a store value -->
  <action name="wcfAutoSetValueAction">
    <param name="valueKey" value="storeIdentifier"/>
    <param name="value" value="AdvancedB2BDirect"/>
  </action>

  <!-- Run Catalog tool testcase -->
  <action name="wcfAutoRunTestCaseAction">
    <param name="url" value="/testdata/commerce/catalog/restricted/test.xml"/>
  </action>

  <!-- Run Marketing tool testcase -->
  <action name="wcfAutoRunTestCaseAction">
    <param name="url" value="/testdata/commerce/marketing/restricted/test.xml"/>
  </action>


```

Define a variable

Run a test case

Test cases are made up of a series of actions described in XML. Actions are run sequentially and one test case can launch another test case. This code sample is from the main test.xml file. First a variable is created and set to the store identifier for this test. Next, the test script invokes tool-specific test cases. The `wcfAutoRunTestCaseAction` has the effect of including the test case inline in the current file. One test case must finish before the next can begin.

Test case structure

```

<testcase>
  <!-- Create a new product -->
  <action name="wcfAutoCreateNewObjectAction">
    <param name="objectType" value="Product"/>
    <param name="objectKey" value="product"/>
  </action>

  <!-- Generate a part number value -->
  <action name="wcfAutoSetValueAction">
    <param name="valueKey" value="code"/>
    <param name="valuePrefix" value="test"/>
  </action>

  <!-- Set the part number property -->
  <action name="wcfAutoSetPropertyAction">
    <param name="object" value="product"/>
    <param name="propertyName" value="partnumber"/>
    <param name="valueKey" value="code"/>
  </action>

  <!-- Create an offer price for the new product -->
  <action name="wcfAutoCreateNewObjectAction">
    <param name="parentObject" value="product"/>
    <param name="objectType" value="CatalogEntryOffer"/>
    <param name="objectKey" value="catalogEntryOffer"/>
  </action>

```

Callouts in the diagram:

- Create a primary object (points to the first `<action>`)
- Define a variable (points to the second `<action>`)
- Set a property value (points to the third `<action>`)
- Use a variable (points to the `value="code"/>` parameter in the third `<action>`)
- Create a child object (points to the fourth `<action>`)

This code sample is from the catalog test case. It shows several of the framework actions available to define a test case. First a new product primary object is created. Next, a variable is defined that will add the prefix 'test' to each auto-generated part number. The third action sets the **partnumber** of the product object using the variable prefix defined above. The final action shown here creates an offer object as a child of the product object.

Action-based framework

- Run a test case
- Open a tool
- Create, delete or find an object
- Set a property
- Load child objects
- Save
- Refresh
- Verify object
- More...

The Management Center test automation framework contains many pre-defined actions for creating test cases. A selection of the available actions are shown here. For complete information on actions and how to use each, see the Information Center. A link is provided on the references page of this presentation.

Test output

- Management Center message console
- WebSphere Commerce Developer console
- Trace.log

The output of the Management Center test cases is written to the WebSphere Commerce Developer console and to the trace.log file. An error in a test case causes the test to terminate. The Management Center message console also contains a list of the completed test actions.

Summary

- Development environment updates
- Web services development
- Management Center testing

This presentation began with an overview of development environment updates and then discussed changes to Web services development in more detail. The second half of the presentation focused on Management Center testing.

References

- Working with WebSphere Commerce services

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.component-services.doc/tasks/twvworkingwithservices.htm>

- Tutorial – Creating a WC service module

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.component-services.doc/tutorial/twvstoresoi.htm>

- Management Center test automation framework

http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.management-center_customization.doc/refs/rfttestframework.htm

This slide contains some useful references for using development and test tools in version 7.



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 e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WCS70_DevAndTestTools.ppt

This module is also available in PDF format at: [../WCS70_DevAndTestTools.pdf](..WCS70_DevAndTestTools.pdf)

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



Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Rational WebSphere

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

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

Other company, product, or service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.

© Copyright International Business Machines Corporation 2010. All rights reserved.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.