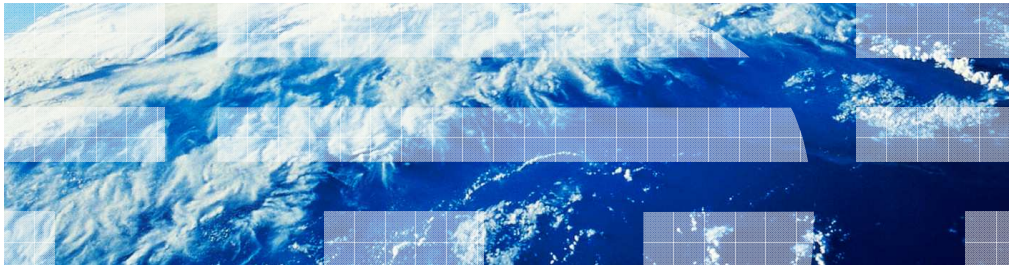


WebSphere Commerce V7.0

WebSphere Commerce Build and Deploy tool



This presentation introduces the WebSphere® Commerce Build and Deploy (WCBD) tool for WebSphere Commerce version 7.

Goals

- To understand the structure of the WCBD tool
- To understand the basics of using WCBD to build and deploy customized code

After viewing this presentation, you should understand the structure of the WCBD tool and how you can use it to build and deploy customized code.

Agenda

- WCBD overview
- Build process
- Deployment process
- Customization highlights
- Best practices

First an overview of the WebSphere Commerce Build and Deploy tool is provided. The two main topics discussed are the build process and the deployment process, followed by customization highlights and best practices.

Overview

- Reference application in version 6
- Shipped with version 7
- Series of Apache Ant scripts
- Customizable

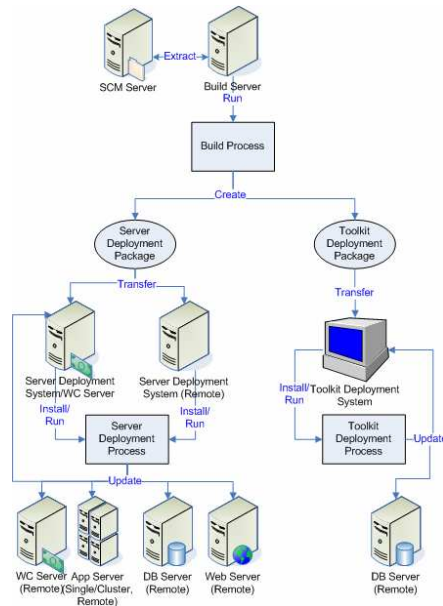
WebSphere Commerce Build and Deploy (WCBD) is the first tool to be developed by the WebSphere Commerce development team that addresses the build and deployment aspect of a WebSphere Commerce customization project. The tool provides a standardized, ANT-based, build and deployment framework, and is designed based on the best practices documented in the WebSphere Commerce Information Center. WCBD is designed to satisfy most typical build and deployment needs, leaving only a few configuration steps for a quick start. When needs differ from the standard implementation, the framework scripts can be customized.

Supports

- CVS, Subversion and ClearCase®
- SQL, mass loadable XML, access control load files
- Deployment to multiple environments
- FTP and SCP/SFTP
- E-mail notification upon build/deployment success/failure

Much of the flexibility of WCBD comes from the many tools it supports as default configurations. The build process works with source configuration management (SCM) systems including CVS, Subversion and Rational® ClearCase. It can be extended to support other SCM systems. It supports deployment of customized database assets using SQL files and XML files that work with the WebSphere Commerce loading utilities. It supports deployment of Java™ EE assets to a WebSphere Application Server using the wsadmin tool. Rollout update for clustered environments is also supported. It supports deployment of Web server assets using FTP, SCP or SFTP, and can be extended to support other transfer methods. Once built, packages can be repeatedly deployed to multiple target environments such as your production, staging, testing, sandbox, and development environments. The tool provides logging and e-mail notification functionality, which helps in problem determination. The tool is highly customizable using Apache Ant scripts to suit project-specific needs. The default scripts are modular, and can be reused to save time when customizing your build process.

System flow



- Three main processes:
 1. Build
 2. Server deployment
 3. Toolkit deployment

6

WebSphere Commerce Build and Deploy tool

© 2010 IBM Corporation

The diagram above shows three main processes supported by WCBD. On the build server, the build process is run. It extracts source code from an SCM system, performs build-related tasks such as compilation and EJB deployment, and creates two deployment packages. One package is for WebSphere Commerce Server environments and the other is for WebSphere Commerce Developer environments. These packages are used to deploy the customized assets. On the server deployment system, the server deployment package is installed, configured and run. It deploys customized assets to the database, the application server (specifically the WebSphere Commerce Enterprise Application), the Web server, and the WebSphere Commerce server. On the toolkit deployment system, which is the machine with WebSphere Commerce Developer V7.0 installed, the toolkit deployment package is installed, configured and run. It deploys customized assets to the database and the toolkit workspace.

Knowledge prerequisites

- Source configuration management system
- Remote transfer protocol (FTP, SCP/SFTP)
- The overall setup for your customization project
- Deployment of customized WebSphere Commerce assets
- Apache Ant
- WebSphere Application Server deployment using Jython scripting

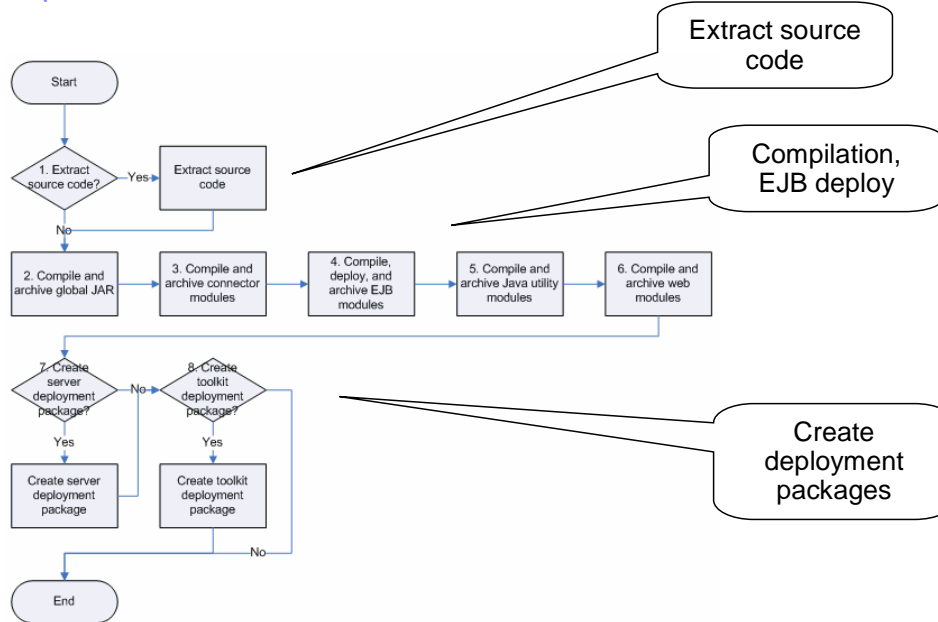
Before beginning the build and deployment process, there are several things you must be familiar with. Many of these prerequisites are related to your specific environment including the SCM system being used, the required remote transfer protocol and the overall setup of your WebSphere Commerce customization project. You should also be familiar with how custom WebSphere Commerce assets such as configuration files, database scripts and EAR files are deployed. Finally, you need to know how to create Apache Ant scripts and possibly Jython scripts. The Jython scripts are specific to WebSphere Application Server deployment.

Section

Build process

This section introduces the build process.

Build process flow



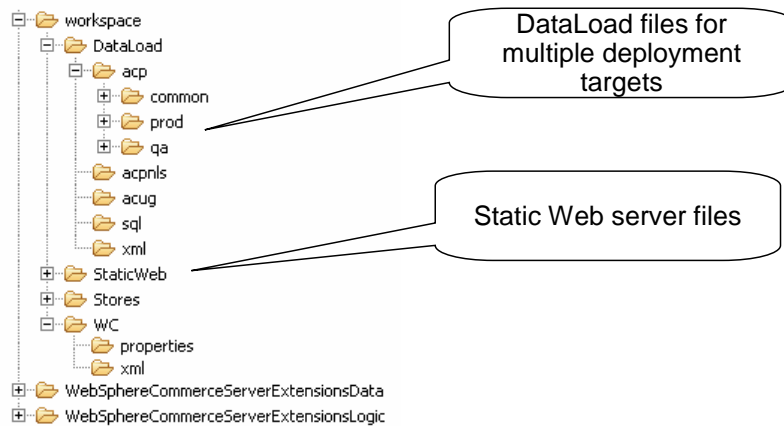
This slide shows a diagram of the build process flow. First, the source code is extracted. Next, a global JAR is created that is used to compile each module individually. This is to help resolve class path circular dependencies during the compilation phase. For each module that is set to be built, the code is compiled and an archive file created. Enterprise beans, if any, are also deployed. Finally, the server and toolkit deployment packages are created.

Configuring and running the build process

1. Set up repository structure
2. Copy WebSphere Commerce libraries to build system
3. Set up source extraction Ant script
4. Configure build settings
5. Run build process

Configuring and running the build process can be divided into five main steps. Step 1, setting up the repository structure, is discussed on the next slide. Running the build process does not require WebSphere Commerce be installed on the build system but some WebSphere Commerce libraries are required for compilation. In step two, an Ant script is provided that can be run on a system with WebSphere Commerce installed. The script creates a compressed file of the required libraries which can then be transferred back to the build machine. Steps three (extracting source code), four (configuring build settings) and five (running the build process) are discussed later in this section.

Repository structure



The Build and Deployment tool is configured by default to use a specific reference repository structure. The structure reflects the WebSphere Commerce Developer installation directory structure. This provides an intuitive code check-in structure for your development team. If you are unable to use the default repository structure, the highly configurable nature of the Build and Deployment tool makes it possible to change the configuration of the build process to adapt to your own project structure. The workspace folder contains projects that are part of the WebSphere Commerce Developer workspace. Some common projects that are likely to be modified in a customization project include Stores, WebSphereCommerceServerExtensionsData and WebSphereCommerceServerExtensionsLogic. The Data load project holds files of various formats that are used to load data during deployment. Each data load subdirectory contains a common directory and other directories specific to deployment targets. For example, the prod and qa directories are shown in the diagram.


The StaticWeb project holds static Web server assets. This project can be left empty if there are no static Web server assets involved in the customization project.

Source extraction Ant script

- Extracts files from a source code repository
- Samples provided for
 - CVS
 - ClearCase
 - Subversion
 - Local file system

To support the use of different SCM systems in different customization projects, the Build and Deployment Tool requires you to provide a custom source extraction Ant build file that is run as part of the build process. Sample scripts for some popular SCMs such as CVS, ClearCase and Subversion are provided. There is also a sample script for extracting files from a local file system.

Build settings

- Templates provided
- Environment variables
 - setenv.bat
- Security sensitive properties
 - build-private.properties
- Build properties 
 - build.properties

There are three files you need to complete to configure your build process. Templates are provided for each of the files. To begin your configuration, make a copy of the template, rename it and fill in the properties for your customization project. In the setenv file, update the path information such as the location of Ant and Java on your build system. Complete the build-private.properties file according to the comments in the file. Properties in this file are encoded when the build process is run. Complete the build.properties file according to the comments in the file. This file is long but well documented. Make sure you read and understand it thoroughly. It is the key to your build process running successfully. You can find a list of the minimum properties that need to be set in the WebSphere Commerce Information Center. See the References page of this presentation for a link.

Build process

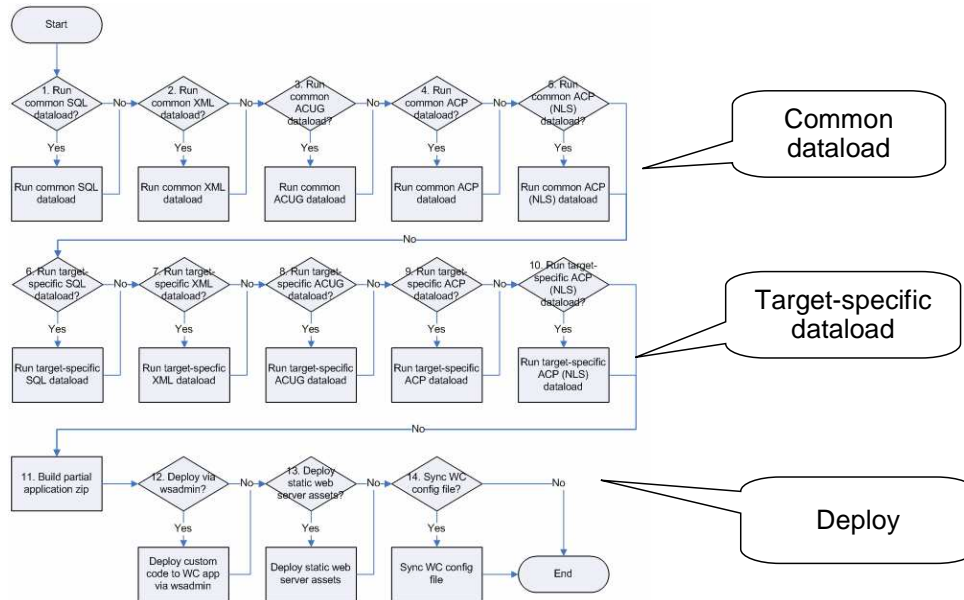
- wcbd-ant.bat -buildfile wcbd-build.xml
-Dbuild.label=<build identifier>
- Log file
– <Install dir>\wcbd\logs\<build identifier>-<timestamp>
- Build success/failure e-mail sent if configured
- Output
 - Server deployment file
 - Toolkit deployment file

Once all configuration steps are complete, you can run your build process to create deployment packages for your target environment. The build label you provide when you run the build process is used to identify the deployment packages and the build log file. The date of when the build is performed or the tag/branch name of the SCM are some good examples of a build label. In addition to providing runtime status in the console, the build process generates a log file that you can find in the directory shown on the slide. The build process can also be configured to send out e-mail notifications upon success or failure. In a build failure notification, the log files are archived and attached for your convenience. Upon completion, the server and toolkit deployment packages are created. These packages can then be installed and configured for deployment as described in the next section.

Deployment process

This section covers the WebSphere Commerce server and toolkit deployment processes.

Server deployment process flow



16

WebSphere Commerce Build and Deploy tool

© 2010 IBM Corporation

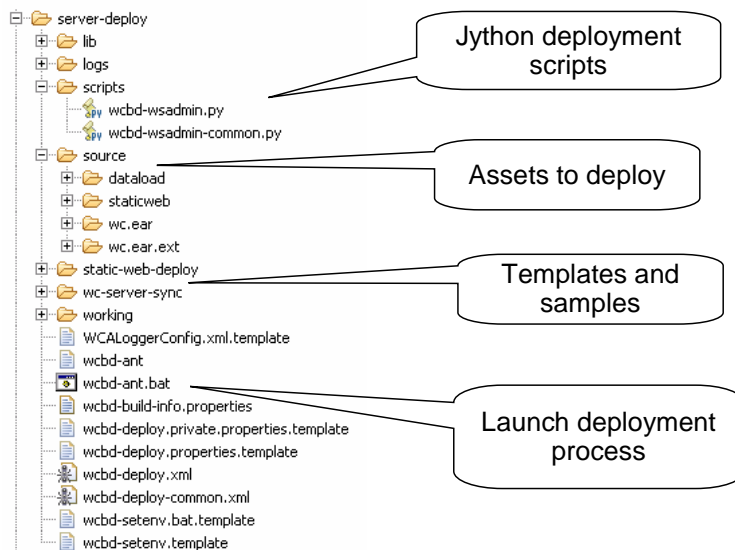
Once you have created the server deployment package, you can deploy it using additional Ant scripts provided as part of the WebSphere Commerce Build and Deployment tool. The high level steps in the server deployment process are shown here. First the database assets are deployed. Database assets can have many forms and each is applied in turn. The common files are applied first, followed by any target specific files. Next the EAR assets are deployed to WebSphere Application Server. You can deploy to either a single server or a cluster. Following the EAR deployment, static Web server assets are copied to the Web server. The final step is to synchronize the WebSphere Commerce configuration file.

Configuring and running the server deployment process

1. Install server deployment package
2. Set up static Web server assets deployment Ant script
3. Set up WebSphere Commerce configuration file synchronization Ant script
4. Configure server deployment settings
5. Run server deployment process

Configuring and running the deployment process can be divided into five main steps. The first step is to install the deployment package. You do this by unzipping the deployment file. The second step is to set up the static Web server assets deployment Ant script. Similar to the source extraction Ant script in the build process, you provides an Ant script that remotely connects to the Web server to deploy assets. Sample scripts are provided for FTP, SCP/SFTP and the local file system. The third step is to set up an Ant script to synchronize the WebSphere Commerce configuration file. This script ensures the master copy of the file on the WebSphere Commerce server is synchronized with the copy in the WebSphere Commerce EAR. Steps four (configuring deployment settings) and five (running the server deployment process) are discussed later in this section.

Server deployment package structure




18

WebSphere Commerce Build and Deploy tool

© 2010 IBM Corporation

This slide shows the structure of the server deployment package created by the WCBD build process. The lib directory contains the Java libraries required by the deployment process. The scripts directory contains the wsadmin Jython deployment scripts. These include the default Jython script that controls the wsadmin deployment process to the WebSphere Commerce EAR and the Jython script that contains reusable methods for the wsadmin deployment process. The source directory contains the assets to deploy. The static-web-deploy directory contains templates and samples for static Web assets deployment Ant scripts. These are the scripts mentioned in step two on the previous slide. The wc-server-sync directory contains templates and samples for WebSphere Commerce configuration file synchronization Ant scripts. These are the scripts mentioned in step three on the previous slide. The working directory contains intermediary or temporary assets generated by the deployment process. wcbd-ant.bat is the batch file (or shell script) for launching the deployment process. You can find more information on each of the configuration files in the Information Center.

Configure server deployment settings

- Templates provided
- Environment variables
 - setenv.bat
- Security sensitive properties
 - deploy-*<target-env>*-private.properties
- Deployment properties
 - deploy-*<target-env>*.properties 

Similar to the task of configuring the build settings, configuring the server deployment settings is the most important step in the deployment process. It can also be the most time consuming because of the many configurations the Build and Deployment Tool supports. There are three files you need to complete to configure your server deployment process. Templates are provided for each of the files. To begin your configuration, make a copy of the template, rename it and fill in the required properties. In the setenv file, update the path information such as the location of Ant and Java on your deployment system. Complete the private deployment properties file for your target environment according to the comments in the file. Properties in this file are encoded when the deployment process is run. Complete the general deployment properties file for your target environment according to the comments in the file. This file is long but well documented. Make sure you read and understand it thoroughly. It is the key to your deployment process running successfully. You can find a list of the minimum properties that need to be set in the WebSphere Commerce Information Center.

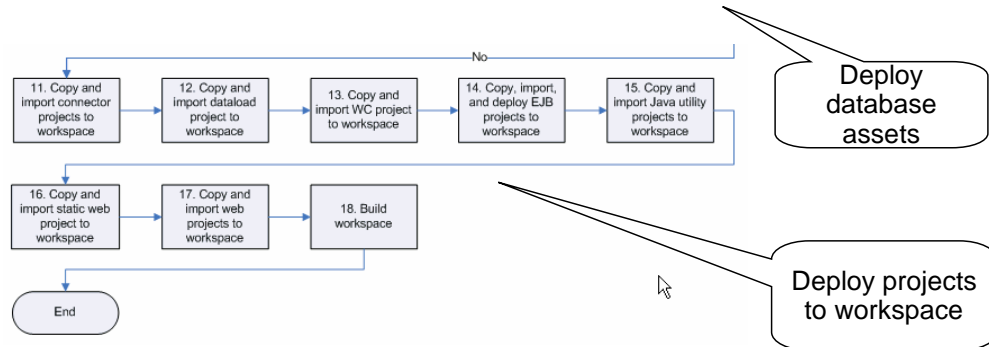
Server deployment process

- Backup
 - Database
 - WebSphere Commerce EAR file
 - WebSphere Application Server configuration files
- `wcbd-ant.bat -buildfile wcbd-deploy.xml -Dtarget.env=target-env`
- Log file
 - `<deploy_server_dir>/logs/<build-label>-<target-env>-tstamp/wcbd-deploy-server.log`
- Deploy success/failure e-mail sent if configured

Before running the server deployment process, it is very important that you backup your existing target system. This includes making a backup of the database, the WebSphere Commerce EAR file and WebSphere Application Server administrative configuration files. If you do not make these backups, you cannot recover if a deployment failure occurs.


Once your existing system is backed up, you can launch the deployment process by running the command shown here, where *target-env* is the identifier for the target environment. Deployment information is output to the console and written to a log file at the location shown on the slide. The server deployment process can be configured to send out e-mail notifications upon success or failure.

Toolkit deployment process flow



This diagram shows a portion of the toolkit deployment process. Not shown here are the steps for deploying database assets. This is the same as the process diagram you saw for server deployment. What differs in toolkit deployment is the latter part of the process where projects for each asset type are deployed to a workspace rather than assets being deployed to WebSphere Application Server.

Configuring and running the toolkit deployment process

1. Install toolkit deployment package
 2. Configure toolkit deployment settings 
 3. Run toolkit deployment process
- Setup steps very similar to server deployment

One of the benefits of using WCBD is the similar structure it brings to each new process. Configuring and running the toolkit deployment process is very similar to the server deployment process even though the target environments are quite different. The first step is to install the deployment package which you do by unzipping the file. The next step is to complete all your deployment configuration files. The same three files you saw in the build process and server deployment process also apply to the toolkit deployment process. The final step is to launch the deployment process. More detail on this step is provided on the next slide.

Toolkit deployment process

- Backup
 - Database
 - Workspace
- `wcbd-rad-ant.bat -buildfile wcbd-deploy.xml -Dtarget.env=target-env`
- Republish WebSphere Commerce application
- Log file
 - `<deploy_toolkit_dir>/logs/<build-label>-<target-env>-tstamp/wcbd-deploy-toolkit.log`

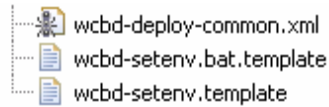
Before running the toolkit deployment process, it is very important that you backup your existing development system. This includes making a backup of the database and the Rational Application Developer workspace. If you do not make these backups, you cannot recover if a deployment failure occurs. Once your existing system is backed up, you can launch the deployment process by running the command shown here, where *target-env* is the identifier for the target environment. Once the deployment process is complete, you need to republish your WebSphere Commerce application. Deployment information is output to the console and written to a log file at the location shown on the slide.

Customization and best practices

This section introduces customization options and best practices for working with WCBD.

Customization highlights

- Files with wcbd prefix are not customizable
- Files with –common in their name are reusable and provide customization examples
- Template files have documentation on customization



- Use a <project>- prefix to identify custom files
- See WCBD documentation for detailed steps

All files that start with the prefix wcbd- are deemed non-customizable for serviceability reasons. You should not modify these files directly. The Build and Deployment Tool provides helper assets that are reusable and serve as examples for customization. Such assets are generally marked using -common in the file name and are very well documented within the file. Always examine these files before you start customizing to understand what can be reused. Template files, identified by the .template file suffix, have sections identified by the string CUSTOMIZATION which provide details and tips on customizing files created from them. The screen capture on this slide shows an example of each of these naming conventions. When creating customized files, use a prefix *project-*, where *project* is an identifier of the project. For example, if the customization project is called "abc" and a customized version of wcbd-build.xml is required, the suggested name of the new file is abc-build.xml.

You can find detailed information on WCBD customization in the Information Center.

Customization examples

- Other SCMs such as CMVC and SourceSafe
- Unit test and code validations
- Custom processes such as DB backup images
- WebSphere Application Server settings such as EJB references and JVM settings
- All three processes can be customized by plugging in custom scripts
 - Ant scripts
 - Wsadmin Jython scripts

WCBD suits most needs through its existing configuration options, but each project might have very specific needs that a generic tool cannot support. This slide lists some examples of when WCBD customization might be needed. The build, server deployment, and toolkit deployment processes can each be customized by plugging in custom scripts. Most of the time you will create custom Ant scripts but custom wsadmin Jython scripts can also be created.

Best practices

- Include template deployment files in build process
- Backup files before deployment
- Check configuration file options before customizing

Once you have gone through the process of creating your deployment configuration files, you can save that work by checking the files into your SCM system so they are extracted during the build process. This greatly simplifies your deployment process since you don't need to re-create your deployment properties each time. This technique applies to static Web asset deployment Ant scripts, WebSphere Commerce configuration file synchronization Ant scripts and target-specific properties files. You should always follow the recommended backup measures before beginning a deployment. Not taking recovery measures will result in not being able to recover if a deployment failure occurs. Always check the configuration files from the Build and Deployment Tool first to see if the customization can be done with default functionality. The configuration files are designed to handle most configuration options such as changing directories and changing the settings of various tools. This will avoid duplicated work.

Limitations

- V7 data load tool not supported

Currently the V7 data load utility is not supported by WCBD. If you want to run the data load utility as part of your deployment process it can be added as a customization.

Summary

- WCBD overview
- Build process
- Deployment process
- Customization highlights
- Best practices

This presentation began with an overview of the WebSphere Commerce Build and Deploy tool. The two main topics covered were the build process and the deployment process. The presentation concluded with customization highlights and best practices.

References

- WCBD documentation

<http://publib.boulder.ibm.com/infocenter/wchelp/v7r0m0/index.jsp?topic=/com.ibm.commerce.developer.doc/concepts/cdewcbdooverview.htm>

This slide contains some useful references for the WebSphere Commerce Build and Deploy tool 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_BuildAndDeployTools.ppt

This module is also available in PDF format at: [../WCS70_BuildAndDeployTools.pdf](..\\WCS70_BuildAndDeployTools.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:

ClearCase 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>

Rational is a trademark of International Business Machines Corporation and Rational Software Corporation in the United States, Other Countries, or both.

Dataload, EJB, Java, JVM, 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.