

# WebSphere Commerce V7 Fix Pack 3

## WebSphere Commerce Build and Deployment enhancements



This presentation provides an overview of the enhancements to the WebSphere Commerce Build and Deployment tool in WebSphere Commerce Version 7 fix pack 3.

---

## Table of contents

- WebSphere Commerce Build and Deployment overview
- Build process
- Deployment process
- Delta deployment process

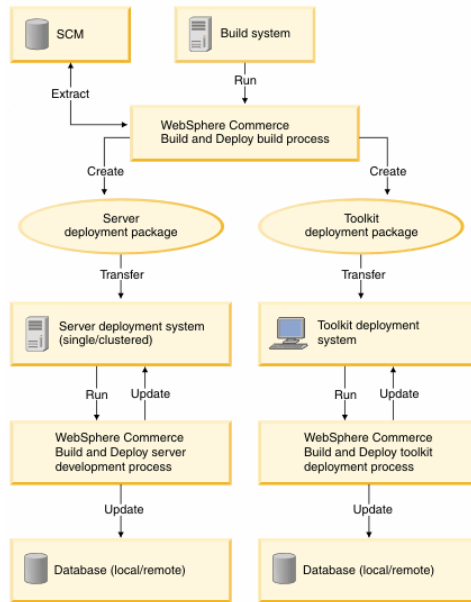
This presentation begins with an overview of the WebSphere Commerce Build and Deployment tool (WCBD). Following the overview is a review of the standard build and deployment processes. The final section of this presentation introduces the new delta deployment process.

Section

## ***Overview***

This section introduces the WCBD tool.

## System flow



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

The WebSphere Commerce Build and Deployment tool provides a standardized, ANT-based framework for building and deploying custom assets. This diagram shows the three main processes supported by WCBD. On the build server, the build process is run. It extracts source code from a source configuration management (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 WebSphere Commerce enterprise application, and the web server. On the toolkit deployment system, the toolkit deployment package is installed, configured and run. It deploys customized assets to the database and the WebSphere Commerce Developer 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.

## ***Build process***

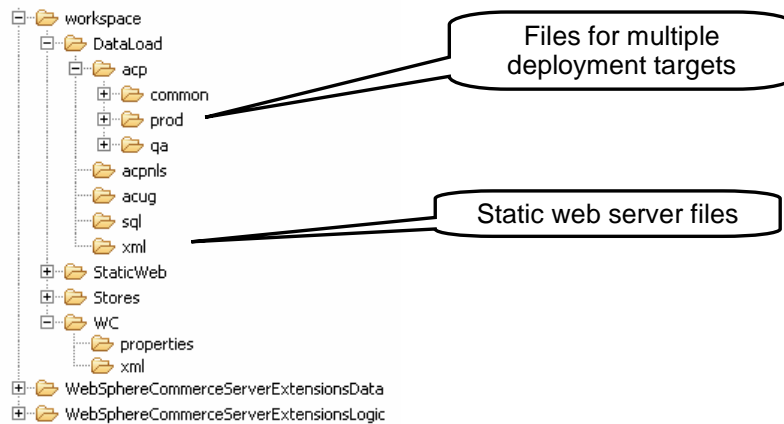
This section reviews the build process.

## Configuring and running the build process

1. Set up repository structure
2. Set up source extraction Ant script
3. Configure build settings
4. Run build process

Configuring and running the build process can be divided into four main steps. Step 1, setting up the repository structure, is discussed on the next slide. Step two (extracting source code), three (configuring build settings) and four (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 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 DataLoad project holds files of various formats that are used to load data during deployment. Each DataLoad 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.



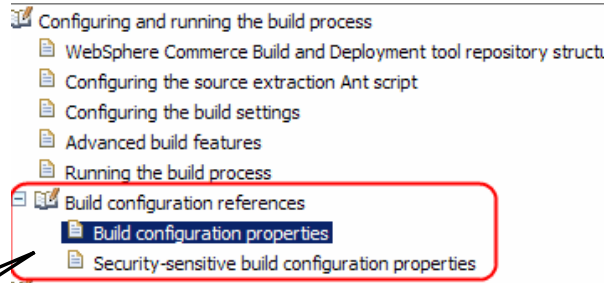
## Source extraction Ant script

- 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



New reference pages for build properties

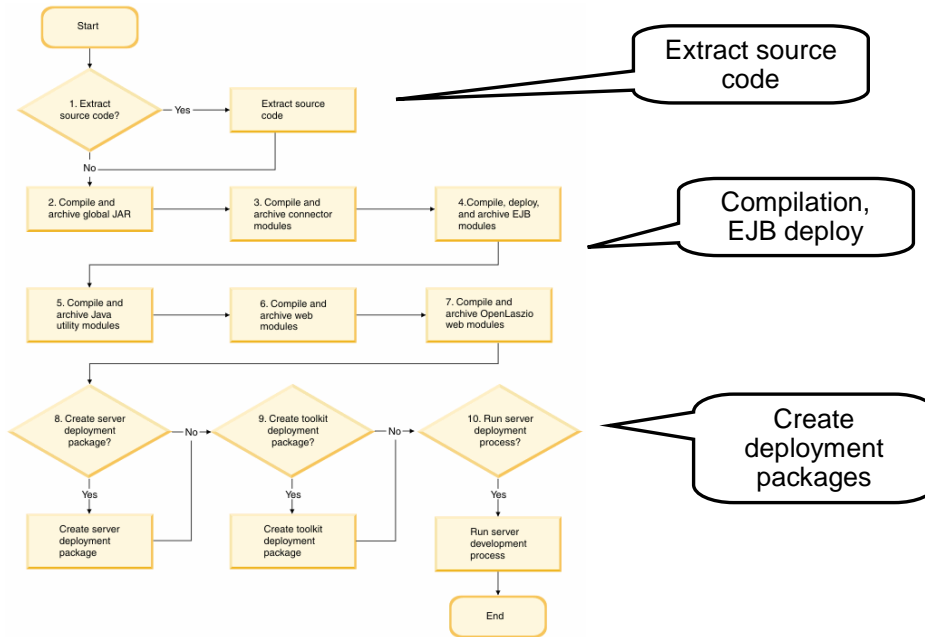
There are three files you need to complete to configure your build process. Templates are provided for each of the files. To begin your configuration, make a copy of the template, rename it and fill in the properties for your customization project. In the setenv file, update the path information such as the location of Ant and Java on your build system. Complete the build-private.properties file with the mail system user name and password. Properties in this file are encoded when the build process is run. Complete the build.properties file to define the build settings and modules to be included in the deployment packages. You can find a build configuration properties reference section in the Information Center, this is new in fix pack 3. 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 email sent if configured
- Output
  - Server deployment package
  - Toolkit deployment package

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 email 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.

## Build process flow



12

WebSphere Commerce Build and Deployment enhancements

© 2011 IBM Corporation

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 in the build properties file, 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.

## ***Deployment process***

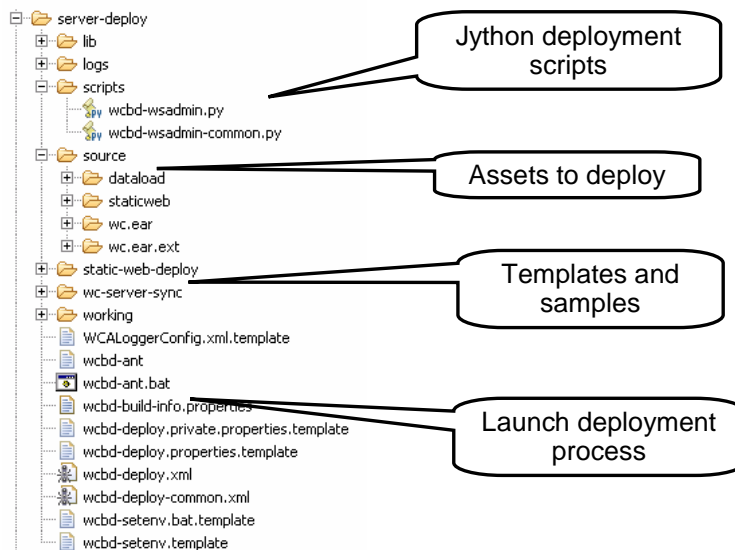
This section reviews the server deployment process.

## Configuring and running the server deployment process

1. Unzip 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 extract the deployment package. 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 provide 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



15

WebSphere Commerce Build and Deployment enhancements

© 2011 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 asset 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

- Configuring and running the server deployment process
  - Installing the server deployment package
  - File structure of the server deployment directory
  - Setting up the static Web server assets deployment Ant script
  - Setting up the WebSphere Commerce configuration file synchronization
  - Setting up the data load utility customization deployment Ant script
  - Configuring the server deployment settings
  - Advanced server deployment features
  - Running the server deployment process
- Server deployment configuration references
  - Server deployment configuration properties
  - Security-sensitive server deployment configuration properties

New reference pages  
for server deployment  
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 with the IDs to connect to WebSphere Application server, your database server and your mail server. Properties in this file are encoded when the deployment process is run. Complete the general deployment properties file for your target environment with the environment details and modules to deploy. This file is long but well documented. Make sure you read and understand it thoroughly. You can find a server deployment configuration properties reference section in the Information Center, this is new in fix pack 3. See the References page of this presentation for a link.



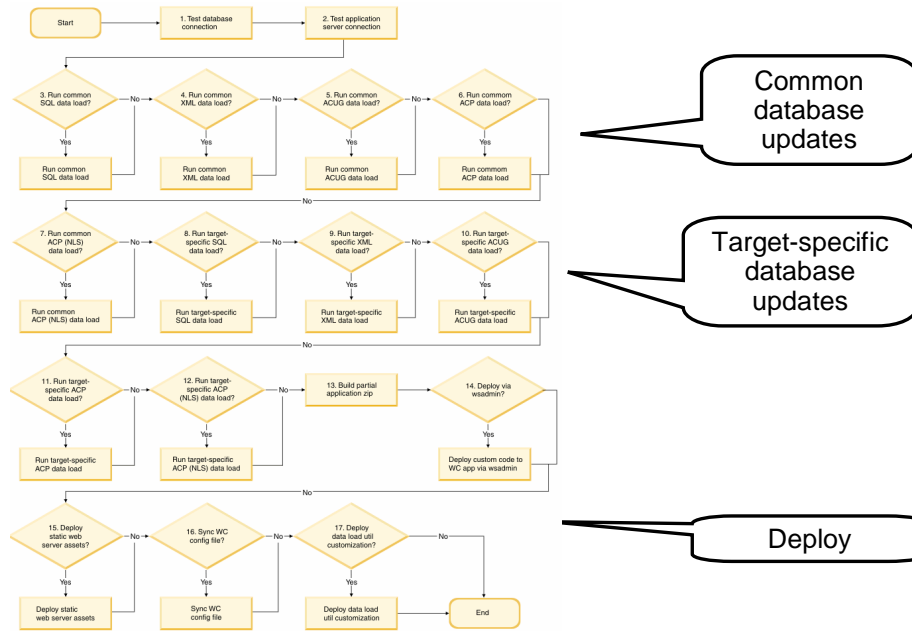
## Server deployment process

- 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 email 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 email notifications upon success or failure.

## Server deployment process flow



18

WebSphere Commerce Build and Deployment enhancements

© 2011 IBM Corporation

This slide shows the flow of the server deployment process. Once you have copied the server deployment package onto your target deployment server, you can deploy it using additional Ant scripts provided as part of the WebSphere Commerce Build and Deployment tool. 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.

## ***Delta deployment***

This section introduces the new delta deployment process.

## What is delta deployment?

- A way to apply incremental updates after initial deployment
- Ideal for updates that affect a small subset of your customized assets
- Can reduce total deployment time

<b>Baseline</b>	<b>Target</b>	<b>Delta</b>
• EJB project 1	• EJB project 1	• EJB project 2'
• EJB project 2	• EJB project 2'	• Stores'
• Stores	• Stores'	• Web project 1
• Java project 1	• Web project 1	
• LOBTools project	• Java project 1	
	• LOBTools project	

In fix pack 3, WCBD has been enhanced to include a delta deployment option. This new capability allows you to deploy only the assets that you have changed since the last deployment. This can make your deployment time much shorter, especially for small or localized updates. For large updates that affect most or all of your deployed assets, you should continue to use the regular deployment option.

Delta deployment works by comparing your current deployment package, called the target, to the most recently deployed package, called the baseline. Modules that have been changed or added since the last deployment are identified and only these modules are deployed. In the case of web modules, if the module has been deployed before only the modified files are deployed by delta deployment.

Delta deployment can be run as a single step or two step process. In the two step process, you first compare the deployment packages and then manually launch the delta deployment. With the one step process, the comparison and deployment are run in sequence which allows the delta deployment process to be automated.

## How does delta deployment work?

- Starting point
  - Baseline deployment package (already deployed to server)
  - Target deployment package (output of most recent build)
- Delta preparation and deployment process
  - Target package is compared to baseline package
  - Delta deployment directory and properties files are generated
  - Modified assets are deployed using standard deployment process with delta flag
- End point
  - Target server has been updated
  - Target deployment package becomes baseline for next update

Since delta deployment is based on comparing the contents of deployment files, you must have completed at least one server deployment using WCBD before you can use the delta deploy option. When you have new custom code to deploy, you run the standard build process to create a new deployment file and fully configure the deployment properties for your target server. Rather than running the standard deployment process at this point, you run a delta preparation process that compares the target package to the baseline package. Any new or modified assets are placed in the delta deployment directory and a delta deployment properties file is automatically generated. Next, the deployment process is run with a delta flag that signals the delta deployment properties file should be used for the deployment instead of the original full deployment properties file. Once the deployment is complete, your target server is up to date and your target deployment package becomes the baseline for the next update.

## Assets included in delta deployment

Module type	Included in delta update	Configuration property name
Connector	Modified connector modules	connector.module.list
EJB	Modified EJB modules	ejb.module.list
New web module	All new web modules	new.web.module.list
Existing web module	Modified files	existing.web.module.list
All other files: EAR files, data files, static web files	Modified files	

This table summarizes what assets are deployed when differences are found between the target and baseline deployment packages. For connector and EJB modules the entire module is redeployed if changes are found. If new web modules are present then the entire web module is deployed. For existing web modules and other asset types, only the individual modified files are deployed. The third column in the table lists the configuration property name corresponding to the various module types.

## Deletions in delta deployment

- Deleted modules
  - List deleted modules in the deployment properties file
    - connector.module.delete.list
    - ejb.module.delete.list
    - web.module.delete.list
- Deleted files in a web module or other EAR file
  - A partial application delete file is generated by the delta prepare process
    - META-INF/ibm-partialapp-delete.props
- Deleted static web assets
  - A deletion file is generated by the delta prepare process
    - delta-delete-includes.txt
  - Static web assets deployment script updated to handle file deletion during delta deploy

In addition to new and changed files, delta deployment can also delete files and modules that were previously deployed. For connector, EJB and web modules, you must list any modules to be deleted in the deployment properties file. The new property names are shown on the slide. For individual deleted files in web modules or other EAR files, the delta preparation process can identify them automatically. A partial application delete file is generated for use in the delta deployment process. The delta preparation process can also identify deleted static web assets. The deployment script for static web assets has been updated to look for the generated deletion file and to perform the deletion during delta deploy.

## Deletion limitations

- Deletion of targetable files is not supported
- Deployed files that were renamed from a targetable template are not deleted if the targetable file is deleted.
- Example
  - configFile.targetable.dev.xml is deployed as configFile.xml in the dev environment
  - configFile.targetable.dev.xml is deleted in a delta deployment
  - configFile.xml is not deleted in the dev environment

There is one limitation to the deletion of files in the delta deployment process. If you are using targetable files to deploy different versions of a file to different environments, the deployed file is not deleted if the original targetable file is deleted. An example is shown on the slide to illustrate this limitation.



## Delta deployment process

Install server deployment package (target package)  
Set up static web server assets deployment Ant script  
Set up WebSphere Commerce configuration file synchronization Ant script  
Configure server deployment settings

---

Same as  
standard  
deployment

Run delta preparation process  
Run server deployment process with delta flag

**OR**

Run delta deployment process

New for delta  
deployment

This slide shows how the server deployment process shown earlier changes when delta deployment is used. The initial steps are no different. The only change is in the final deployment step. With delta deployment, you have the option to run the comparison and delta deployment as two separate steps or to combine them into a single step. The two step process gives you an opportunity to review the results of the comparison process before the files are deployed. The single step process simplifies automation of the overall deployment process.

## Two step server delta deployment process

- Complete server deployment backup process
- `wcbd-ant.bat -buildfile wcbd-delta-prepare.xml -Dtarget.env=target-env -Dbaseline.dir=baseline_dir`
  - Log file
    - `<WCBD_target_deploy_server_dir>/logs/delta-prepare-<target.env>/wcbd-delta-prepare.log`
  - Also check
    - `<WCBD_target_deploy_server_dir>/delta-summary-<target.env>.xml`
- `wcbd-ant.bat -buildfile wcbd-deploy.xml -Dtarget.env=target-env -Ddelta.mode=true`
  - Log file
    - `<WCBD_target_deploy_server_dir>/logs/<build-label>-<target-env>-tstamp/wcbd-deploy-server.log`
- Deploy success/failure email sent if configured

This slide shows the syntax and log file output for the two step process. With any server deployment, you should begin by following the recommended backup procedures. The delta preparation process accepts the target environment and baseline deployment package as parameters. The deployment process is the same with the addition of the `delta.mode=true` parameter.

## Single step server delta deployment

- Combines prepare and deploy to support automated deployment scenarios
  - Can be set to fail if there are no changes to deploy
- Create baseline.properties configuration file
  - baseline.dir property contains the location of the baseline deployment package
- New command
  - `wcbd-ant -buildfile wcbd-delta-deploy.xml -Dtarget.env=target-env`
  - Log file
    - `<WCBD_target_deploy_server_dir>/logs/delta-prepare-<target-env>/wcbd-delta-deploy.log`
    - `<WCBD_target_deploy_server_dir>/logs/<build-label>-<target-env>-tstamp/wcbd-deploy-server.log`
- baseline.dir property set to `<WCBD_target_deploy_server_dir>` after successful deployment

The single step delta deployment process performs the same tasks but wraps them in a single command. Since there is no manual examination of the comparison results, the command can be set to fail if there are no changes to deploy. The first time you run the single step deployment process, you must provide one new configuration file, baseline.properties. This properties file contains the location of the baseline deployment package. By storing this value in a configuration file, the single step delta deployment process can update the value automatically so it is ready for the next delta deployment. The new command syntax is shown on the slide.

## Summary

- WCBD overview
- Build process
- Deployment process
- Delta deployment process

This presentation began with an overview of the WebSphere Commerce Build and Deployment tool. Following the overview was a review of the standard build and deployment processes. The final section of this presentation introduced the new delta deployment process.

## References

- **WCBD documentation**

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

- **Build configuration reference**

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

- **Server deployment configuration reference**

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

- **Toolkit deployment configuration reference**

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

This slide contains some useful 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\\_WCBD.ppt](mailto:iea@us.ibm.com?subject=Feedback_about_WCBD.ppt)

This module is also available in PDF format at: [../WCBD.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, ClearCase, 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>

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.