



IBM Software Group

WebSphere® Process Server V6.0 WebSphere Integration Developer V6.0

Test Environment



@business on demand.

© 2005 IBM Corporation
Updated December 12, 2005

This presentation will discuss the details of the test environment included with IBM WebSphere Integration Developer V6.0 and support for testing applications on WebSphere Process Server V6.0.

Goals

- Introduce architecture for testing applications in WebSphere Integration Developer
- Understand the server configuration steps and how it is different from previous version
- Describe application and server resource configuration
- Explain additional test environment options



The goals of this presentation include introducing the architecture for testing applications, talking about the different components and functionalities that are involved. Next, the presentation will explain how to set up different server configurations and the tools used to do so, and how the development environment works with the server configurations. Then there is a discussion of application publishing and tools, and some of the additional test environment options that are specific to the WebSphere Process Server.

Agenda

- **Test Environment Architecture**
- Installing Test Environment
- Preparing Test Environment
- Testing Applications
- Working with Server Configuration
- Summary and References



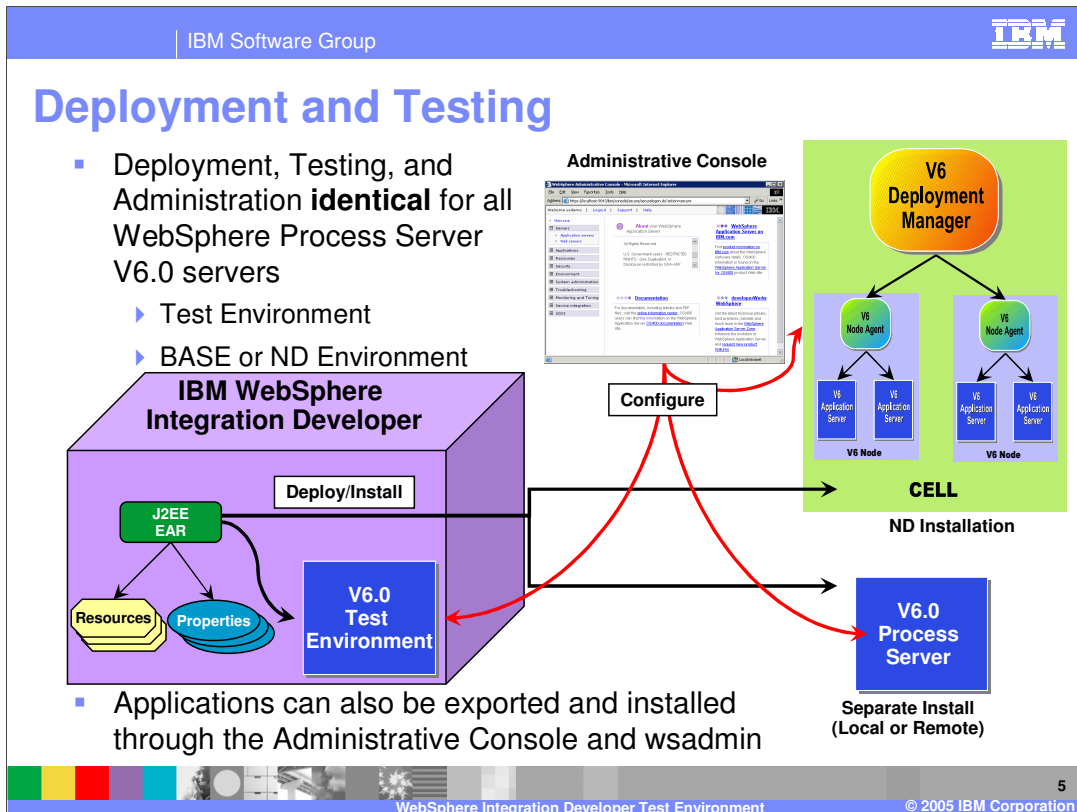
The test environment is a very important piece in the overall development cycle. This section will cover the test environment architecture.

Overview of Test Environment

- Consistent Deployment and Administrative model for local and remote WebSphere® Process Server V6.0
- Architecture to support deploying and testing on WebSphere Process Server V6.0 Base and ND
- Test environment configuration through Administrative Console for WebSphere Process Server V6

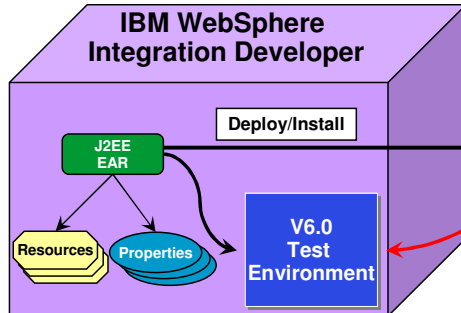


The test environment features a consistent deployment and administrative model for any server you might use – either the application server that is installed as a part of WebSphere Integration Developer, an application server installed on the same physical system, or an application server running remotely, even on another platform. This architecture is an extension of the architecture of WebSphere Application Server Base and Network Deployment versions, and extends the features and functions to include WebSphere Process Server. Unlike earlier versions of the development tools, it is not necessary to configure the test environment through server configurations in the tools; the test environment is configured through the Administrative Console.



- Deployment, Testing, and Administration **identical** for all WebSphere Process Server V6.0 servers

- Test Environment
- BASE or ND Environment



- Applications can also be exported and installed through the Administrative Console and wsadmin

As stated in the previous slide, the deployment and administrative model is identical for all versions of WebSphere Process Server V6. The Integrated Test Environment, a separate installation of WebSphere Process Server on the same system as IBM WebSphere Integration Developer or on a remote system, and even a server which is part of a Network Deployment cell, are all treated the same way for deployment and administration. For deployment, IBM WebSphere Integration Developer allow you to specify a particular WebSphere Process Server, and with a single action applications can be published to the server and started in preparation for testing. If updates are made to an application, the changes can also be scheduled to be published to a server for testing. Applications may also be exported out of IBM WebSphere Integration Developer and be installed using the Administrative Console or the wsadmin command line interface.

As indicated by the diagram, resources and properties for the application can also be created and packaged within an Java™ 2 Enterprise Edition EAR file before it is published.

Agenda

- Test Environment Architecture
- **Installing Test Environment**
- Preparing Test Environment
- Testing Applications
- Working with Server Configuration
- Summary and References

This section covers installing the test environment.

Test Environment Installation

- WebSphere Process Server V6.0 test environment install option in WID
 - ▶ Installed silently
 - ▶ Response files at disk5\external\WAS\
 - ▶ Business Process Choreography, CEI, Application Scheduler
 - ▶ J2C authentication Alias set up with **wid/wid** for username/password
- Install location
 - ▶ Runtime binaries located at `${WID Install Dir}\runtimes\bi_v6`
 - Server Stub directory `${WID Install Dir}\runtimes\bi_v6_stub` is not part of test environment install
 - ▶ Default configuration profile located at `${WID Install Dir}\prof`
 - Profile named WID
- Install results located at `${WID Install Location}\logs`
 - ▶ `wps_v6_install_err.log`
 - ▶ `wps_v6_install_out.log`
 - ▶ Collected from the different WPS log files located at `${WID Install Dir}\runtimes\bi_v6\logs` and more

The WebSphere Process Server test environment installation is not automatic; it is an option you must select when installing the WebSphere Integration Developer – either the silent install, or using the installation wizard. There are no WebSphere Process Server options to configure during the installation. Even though the response files are visible in the Disk5 directory, the response files should not be changed. As part of the WebSphere Process Server installation, several other things are also installed, including the Business Process Choreography, CEI, the Application Scheduler, and the WPCSRDB, which is a database used to hold business rules, events, and relationship information. J2C Authentication aliases for these components are set up with a default User ID and Password of `w-i-d`. It is important to remember this when you turn on security – it will be necessary to change these values, or to create a user ID in your security registry with these values.


The WebSphere Process Server architecture separates the runtime binaries from the server configuration files. These binaries are written in the `runtimes\bi_v6` directory. You will see a `bi_v6_stub` directory; this folder contains the library files used for the different builders and remote deploy features. These are not technically part of the binaries. The configuration files for the profile are located in the `profs` directory. The profile name need not match the directory name – for the WebSphere Integration Developer, the profile name is “WID”. The reason for moving the `Profs` directory up a few levels is because there are Windows™ operating system imposed limits on path length, and if the `profs` directory was under the `runtimes\bi_v6` directory, then there could be problems with installing process choreography or CEI, and potentially with long Java package names or long application names.

The installation logs are written in the `logs` directory. You can also see the logs in the `runtimes\bi_v6\logs` directory; these log entries are echoed to the higher level directory and

IBM Software Group IBM

Test Environment Differences

- Test Environment installation is different from stand-alone WebSphere Process Server installation
 - ▶ Application servers > server1
- Development Mode set on server
 - ▶ Application servers > server1
- Transaction set to not time out (value 0)
 - ▶ Application servers > server1 > Process Definition > Transaction Service
- Integrated Test Client support added to `#{WID_Install_Dir}\runtimes\bi_v6\lib`
 - ▶ CompTestCommon.jar
 - ▶ CompTestController.jar



The screenshot shows the 'Application servers' configuration page for 'server1'. Under the 'General Properties' section, the 'Run in development mode' checkbox is checked and circled in red. Other visible options include 'Parallel start' (checked) and 'Classloader policy' (set to 'Multiple').

WebSphere Integration Developer Test Environment © 2005 IBM Corporation

There are some differences between the test environment installation and a stand-alone WebSphere Process Server installation. These three settings make up the difference between the test environment and the default production server installation.

When you install the test environment, the server runs in development mode by default. You can change this using the check box on the configuration tab in the WebSphere Process Server Administrative Console. Development mode removes some of the strict checks. For example, in a production server, if you have business processes or human tasks in flight, the server will not accept an application update until those instances have been stopped and deleted. In an iterative development environment, this can be very inconvenient. Turning on development mode removes these restrictions.

The second difference is that the transaction timeout value is set to zero, or no time out. This is useful when debugging and stepping through code; the usual timeout value of about two minutes is short enough that transactions frequently time-out during a debugging session and the debugging information is lost; setting transactions to never time out means that they will be there as long as necessary to debug a problem.

There are some additional JAR files that are placed in the lib directory and used by the test controller application that is installed with the integrated test client. If you want to test to a remote server, these files can be copied to that remote server.

Known installation problems and limitations

- Interim Fix 001a should be applied
 - ▶ <http://www-1.ibm.com/support/docview.wss?rs=2307&context=SSQH9M&uid=swg24010506>
- Long installation paths or long hostnames can cause installations (profile creation) to fail
 - ▶ Check %temp%\log.txt
 - ▶ Manually create profile at a shorter path
 - ▶ Problem may also be seen at application install or application update time
- Previous uninstall did not complete fully/successfully
 - ▶ Clean up system
 - http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/index.jsp?topic=/com.ibm.wspins.doc/doc/tins_trouble.html
 - ▶ Check vpd.properties file
 - **Windows™ 2000** – C:\WINNT\vpd.properties
 - **Windows XP / 2003** – C:\WINDOWS\vpd.properties
 - **Linux®** – /root/vpd.properties



There are a few known items with the installation of interest when supporting WebSphere Integration Developer

Cumulative Interim Fix 001a should be applied after installing WebSphere Integration Developer V6.0 and the Test Environment.

Long installation paths can cause the installer to fail; if so, check the log.txt file in the temp directory. You can also have path length problems later on when applications are installed or updated. Long path lengths don't always show up directly as errors; what you will see is an error message about path or file not found, or a class not found when the class is in a long-named package.

If the previous installation did not completely uninstall, another install may report only partial success – for example, the test environment might not install successfully. The link on the slide details how to manually clean up a previous unsuccessful removal. Sometimes, the entries are not removed from the vpd.properties file. If an installation fails, check these locations and edit the vpd.properties file.

Agenda

- Test Environment Architecture
- Installing Test Environment
- **Preparing Test Environment**
- Testing Applications
- Working with Server Configuration
- Summary and References

This section covers preparing the newly installed test environment.

Local Test Environment

- Profile named “WID” is available to all workspaces
 - ▶ Every workspace starts with a Server pointing to server profile “WID”
 - ▶ Server is ready for testing; nothing else needs to be created
- Creating a new Server in WebSphere Integration Developer only creates a new pointer to “default” profile
 - ▶ No configuration files are created

The screenshot illustrates the local test environment setup. On the left, two workspace icons labeled 'workspace1' and 'workspace2' are shown. Red arrows point from these workspaces to two instances of the 'Servers' view in the IDE. Each instance shows a single server entry: 'WebSphere Process Server v6.0' with 'localhost' as the host name. On the right, a file explorer window shows the directory structure of the WebSphere installation. A red circle highlights the 'prof' folder under 'WebSphere\6.0\bin', which is the default profile. Red arrows also point from this 'prof' folder to the server entries in the two workspace views, indicating that both workspaces share this default profile.

In the local test environment that is installed as part of WebSphere Integration Developer there is a default “WID” profile. Each workspace that you start will have a pointer to this profile. The profile is independent of the workspace, and all you have in the workspace is essentially a pointer to the profile. This means you may see applications show up on the test server that are in different workspaces. Developers – think of the test environment as passed by reference rather than passed by value, and the concept should be clear.

To reduce the confusion of having multiple applications in the test environment, you can create multiple test environment configurations.

Creating a New Local Test Environment

- New profile must be created for a new server with Profile Creation Tool first
 - `${WID_Install_Dir}\runtimes\bi_v6\bin\ProfileCreator_wbi\pcatWindows.exe`

The screenshot illustrates the process of creating a new profile and server in WebSphere Integration Developer. It shows a file explorer on the left with the following structure:

- Local Disk (C:)
 - Program Files
 - IBM
 - WebSphere
 - ID
 - 6.0
 - prof

Two screenshots of the 'Servers' console are shown. The top screenshot shows a single server:

Server	Host name	Status	State
WebSphere Process Server v6.0	localhost	Stopped	Synchronized

The bottom screenshot shows two servers:

Server	Host name	Status	State
WebSphere Process Server v6.0	localhost	Stopped	Synchronized
WebSphere Process v6.0 Server @ localhost	localhost	Stopped	Synchronized

Text annotations on the slide include:

- 'Create a new profile with the WebSphere Process Server Profile Creation Tool' (with an arrow pointing to the 'prof' folder).
- 'Create a new Server' (with an arrow pointing to the 'prof2' folder).

At the bottom of the slide, it says 'WebSphere Integration Developer Test Environment' and '© 2005 IBM Corporation'.

This slide illustrates the fact that in order to create a new test server, you must run the Profile Creation tool, which will create a new set of configuration files. The Profile Creation Tool cannot be launched from within WebSphere Integration Developer so you will need to start it separately. New profiles are created with a unique set of ports in order to avoid port conflicts. If you will only start one profile at a time, you can change these ports to match those of the default profile. This will eliminate the need to remember to specify the correct ports for the server you are currently using. If you have a requirement to start multiple servers simultaneously, you will need to take note of the ports used by the intended server.

Managing Local Test Environment

- WebSphere Integration Developer reads local test environment information from WebSphere V6 Integrated Test Environment installation
 - ▶ Workspace not used to hold configuration information
- Installed runtime location and profileRegistry.xml are used to determine which profiles (configurations) are available as test servers
 - ▶ Installed runtime location can be the default (`${WID_INSTALL_DIR}\runtimes\bi_v6`) or separate installation
 - ▶ profileRegistry.xml points to available profiles
 - `${WID_INSTALL_DIR}\runtimes\bi_v6\properties\profileRegistry.xml`
- wasprofile tool available for managing profiles
 - ▶ Create, list, delete, and validate commands available in case profile directories are deleted directly from the file system
 - ▶ Located at `${WID_INSTALL_DIR}\runtimes\bi_v6\bin`



The workspace no longer contains the test environment configuration information. Instead, pointers are created to the test environment, whether it be the instance installed with WebSphere Integration Developer or an external installation of WebSphere Process Server. The profiles available for use as test environments stored within the profileRegistry.xml. The profileRegistry.xml file is managed using the wasprofile tool, which is included as part of WebSphere Integration Developer, however it is not launched from WebSphere Integration Developer. WebSphere Process Server Profile tool calls the wasprofile tool when creating profiles; therefore to list, delete, and validate existing profiles use the wasprofile tool rather than manually editing the profileRegistry.xml file or simply deleting the profile directory. Manual editing could cause unpredictable results due to broken links and dependencies.

Local Test Environment: Separate Install

- Separate installations of WebSphere Process Server V6.0 on the same system as WebSphere Integration Developer can be set as test environments
- Installed Server Runtime Environment must be configured with location of separate installation
 - ▶ Windows > Preferences > Server > Installed Runtimes
- New Server must point to separate install and appropriate profile
 - ▶ Separate installation can have any number of Profiles
 - ▶ Set correct SOAP or RMI connector port



You can also now use a separate installation of WebSphere Process Server as a test environment. This capability is valuable in situations where you have an existing installation with certain patches already installed. Rather than install the patches to the test environment, you can simply configure WebSphere Integration Developer to utilize the existing installation. Before you can use a separate WebSphere Process Server instance as a test server, you must register that instance using the Windows Preferences under Server -> Installed Runtimes. After you specify the installation directory of the separate instance, a list of existing profiles will be displayed when creating a server in WebSphere Integration Developer. Set the SOAP or RMI connector port for the profile you wish to use as your test environment.

IBM Software Group IBM

New Test Environment: Separate Install

Register separate installation location

Create server which points to separate installation location

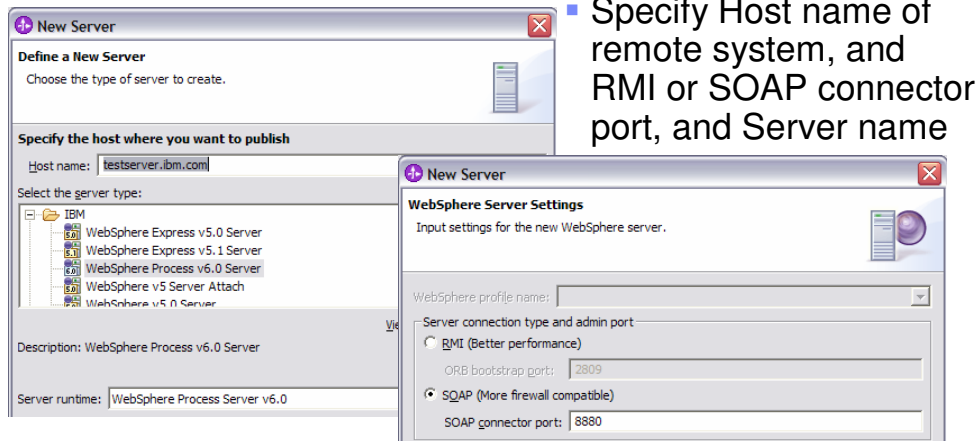
Server	Host name	Status	State
WebSphere Process Server v6.0	localhost	Stopped	Synchronized
WebSphere Process v6.0 Server @ localhost	localhost	Stopped	Synchronized
WebSphere Process v6.0 Server @ localhost - Separate Install	localhost	Stopped	Synchronized

15
WebSphere Integration Developer Test Environment © 2005 IBM Corporation

This slide summarizes the steps necessary for registering a separate installation of WebSphere Process Server on a system and making WebSphere Integration Developer aware of that server. After you register the installation location in the preferences and create a server document in the servers view, you can select that runtime installation location, and then enter the port for the proper server profile.

Remote Test Environment

- Remote installations of WebSphere Process Server V6.0 can also be configured as Test Environments
 - ▶ All platforms



- Specify Host name of remote system, and RMI or SOAP connector port, and Server name

Remote instances of WebSphere Process Server are supported for use as test environments. The remote test environment is very simple to configure. When creating a new server document or pointer document, specify something other than localhost for the Host name and the SOAP connector port for the particular server you wish to connect to.

Preparing to Test on a ND Environment

- Servers which are part of a Cell (ND environment) can be test servers
- Server must be created and pointed to the server
 - ▶ Specify server Deployment Manager SOAP connector port; not server or Node Agent
 - ▶ Specify Network deployment server name
 - ▶ Specify Server type
 - Use Detect capabilities if unsure

New Server

WebSphere Server Settings
Input settings for the new WebSphere server.

WebSphere profile name:

Server connection type and admin port

BMI (Better performance)

ORB bootstrap port:

SOAP (More firewall compatible)

SOAP connector port:

Run server with resources within the workspace

Security is enabled on this server

Current active authentication settings:

User ID:

Password:

Server name:

Server type

BASE, Express or unmanaged Network Deployment server

Network Deployment server

Network Deployment server name:

The server name is in the form of:
<cell name>/<node name>/<server name>
For example, localhost/localhost/server1.

Click this button to detect the server type.

Testing with a server that is part of a Network Deployment cell environment is supported in V6.0 as well. For this type of configuration, you will again specify something other than localhost for the Host name and the SOAP port for the Deployment Manager node. The SOAP connector port for a Deployment Manager can be found in the Administrative console by navigating to System administration -> Deployment manager and then expanding Ports under Additional Properties (SOAP_CONNECTOR_ADDRESS). You also must provide the hierarchical name of the application server, including the cell, node, and server name for the server you are connecting to. The example shown here is somewhat simplified, so be careful to specify the hierarchical name properly. You can determine this by looking at the topology in the Administrative console for the WebSphere Process Server environment. There is also a detect feature to help you determine if the server you are connecting to is a federated member of a Network Deployment cell or a single application server. The correct port number will need to be specified before selecting the Detect button.

Location of a Test Server

- Key fields in the Server can be checked to determine type and location of server
- Values stored in `${workspace_dir}\.metadata\com.ibm.wtp.server.core\servers.xml`

The screenshot shows the 'Server Overview' configuration page for a 'WebSphere Process v6.0 Server @ testserver.ibm.com'. The page is divided into several sections:

- General:** Contains fields for 'Server name' (WebSphere Process v6.0 Server @ testserver), 'Host name' (testserver.ibm.com), and 'Runtime' (WebSphere Process Server v6.0).
- Runtime:** Includes 'Update server status interval (in milliseconds): 5000' and 'Server connection type and admin port' with options for RMI (Better performance) and SOAP (More firewall compatible). The SOAP connector port is set to 8880.
- Security:** Includes 'Enable hot method' and 'debug mode' checkboxes.
- Network Deployment:** Contains 'Server name' (server1) and 'Server type' (Network Deployment server). The 'Network Deployment server name' is set to testserverCell01/testserverNode01/server1.

Yellow callout boxes highlight the following fields:

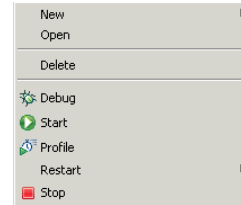
- Check Host name:** Points to the 'Host name' field.
- Runtime – check Preferences for install location:** Points to the 'Runtime' dropdown menu.
- Verify connector port:** Points to the 'SOAP connector port' field.
- Network deployment server:** Points to the 'Network Deployment server name' field.

The footer of the page reads 'WebSphere Integration Developer Test Environment © 2005 IBM Corporation' and the page number '18' is visible in the bottom right corner.

You can obtain the characteristics, including runtime location, server name, and port assignment, for the server you are working with from the server pointer document. This is valuable for verifying that you are connecting to the right server and for troubleshooting any problems. If changes need to be made to the connection you are making to the server, they can be made in this document.

Commands to Manage Servers

- Debug
 - ▶ Only available on local Servers
- Start
 - ▶ Only available on local Servers
- Profile
 - ▶ Only available on local Servers
 - ▶ Requires Remote Agent Controller to be installed on development and deployment systems
- Restart
 - ▶ Available on all active WebSphere V6.0 Servers
 - ▶ Restart in different modes (Normal, Debug, and Profile)
 - ▶ If applications from the workspace are already added to the server, server starts in **debug mode**
- Stop



The types of operations available depend on whether you are using a local test environment such as the default test environment or a separate local install, or a remote test environment. The start operation is only available for local servers, which can be started in either Debug or Profile mode. Both local and remote servers can be stopped and restarted. When a restart command is issued, the last thing that process does is spawn a new process, which causes the restart. This eliminates the need to use the remote agent controller.

Agenda

- Test Environment Architecture
- Installing Test Environment
- Preparing Test Environment
- **Testing Applications**
- Working with Server Configuration
- Summary and References

This section covers testing applications.

Adding Applications to Test Servers

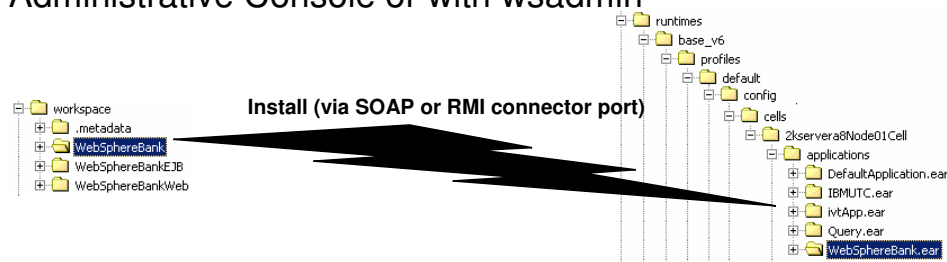
- Run on Server (Debug on Server, Profile on Server)
 - ▶ Select or define new Server for application
 - ▶ Server is started if not already started (local only)
- Add/Remove Projects
 - ▶ Server is started if not already started (local only)
- Publish
 - ▶ Reinstalls applications from the workspace onto the server

Once you have configured application servers, whether they be local or remote, you can run applications on those servers for testing purposes. Applications can be installed onto servers using either the run on server command or the add/remove projects dialog box. Both methods perform the same operation. The server must be started in order to install an application. Both these command will attempt to start a local server if it is not already started. Once the server is started, the application will be installed and started. There are minor differences in behavior between these two commands, but the Add/Remove Projects command is recommended as the best option.

Applications can be reinstalled using the Publish command. This will reinstall all applications that are configured in your workspace.

Adding Applications to Test Servers

- Applications are added through communication through the SOAP or RMI connector port and installed on the server
 - ▶ Applications added to Server configuration and registered in serverindex.xml as part of installation process
 - ▶ Exactly the same as installation through Administrative Console or with wsadmin



22

WebSphere Integration Developer Test Environment

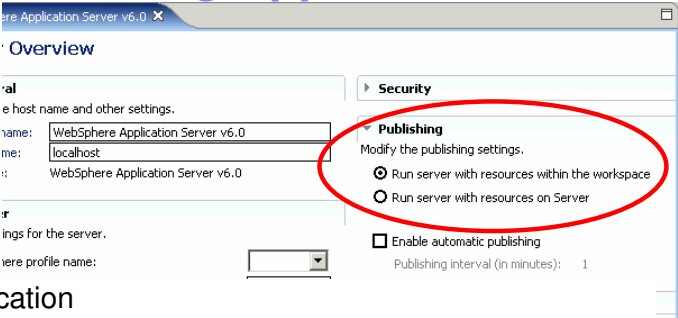
© 2005 IBM Corporation

The call to install the application is made over the SOAP or RMI connector port and the installation takes place just as it would if it were performed from the administrative console or through wsadmin.

IBM Software Group IBM

Consideration for Adding Applications

- When Applications are added to a local test server or separate local installation, the actual binaries can be located in different places
- Application binaries location
 - Workspace (Run server with resources within the workspace)
 - Certain changes made to applications will be picked up automatically
 - Test Server will contain orphaned applications if workspace containing applications deleted
 - Server (Run server with resources on Server)
 - Changes to applications will need to be published
- Setting applies to all applications added from a workspace
- Removal of applications via Add/Remove Projects or the Administrative Console does not remove projects from a workspace



The screenshot shows the 'Overview' page for a WebSphere Application Server v6.0. The 'Publishing' section is expanded, showing two radio button options: 'Run server with resources within the workspace' (which is selected and circled in red) and 'Run server with resources on Server'. There is also a checkbox for 'Enable automatic publishing' and a 'Publishing interval (in minutes): 1'.

23

WebSphere Integration Developer Test Environment © 2005 IBM Corporation

When installing applications onto your test server, you can specify whether the application binaries should be executed out of the workspace or published to the server. Here are some considerations to help you decide works better in your environment:

First, if the application binaries run from the workspace, certain changes, such as saving a .jsp file, will be updated and reloaded automatically. A drawback to this is that if a single test environment is used for multiple workspaces and you deploy multiple applications to that single test server, and then later delete workspaces without removing the application from the server, this will result in error messages during server startup.

Second, whatever settings you choose will apply to all applications for that server.

Third, if the application binaries run on the server, you must republish whenever updates to the applications are made.

Finally, if you change settings after installing an application, the application will continue to run with the old settings. You must remove the application and add it again in order for it to run under the new settings. Applications are removed using the Add/Remove Projects dialog or the Administrative console. When you remove an application using the Administrative Console, it will only be removed from the server configuration, the installation binaries will not be removed from the workspace.

Application Changes (Run from workspace)

- Certain changes will be picked up and take affect without publishing
 - ▶ Changes to servlets, EJBs, and deployment descriptors will be installed/restarted
- For Process Server artifacts (business processes, human tasks, business rules, business state machines, enabling monitoring), certain changes will be picked up and reinstalled
 - ▶ Not guaranteed to pick up changes; issue **Publish** command to be sure
- Server restart required when interface and business objects are updated and republished
 - ▶ Definitions are not completely refreshed with just an application republish/restart
- Other required server restarts
 - ▶ Enabled defined resources
 - Datasources at server level and above, JMS resources
 - ▶ Debugging and Profiling applications
 - ▶ Enabling Security
 - ▶ Changing JVM values on application server

When applications are run from the workspace, which is the default setting, some changes are picked up automatically. Edits to servlets, JSPs, deployment descriptors, and so on are picked up when the code is saved and the updated code is restarted in the local test environment. For most Process Server artifacts, this is the case; however, there are cases where some things are not reinitialized automatically. To force changes to be picked up, use the Publish command to re-install applications completely. If Publishing does not pick up all the changes, then restarting the server will force a complete refresh.

Other situations that require a server restart include enabling some resources, such as data sources, turning on debugging or profiling, turning security on or off, and changing JVM settings on the application server.

Application Changes (Run on Server)

- Changes to installed applications (resources on server) can be published manually (default) or automatically to a test server
 - ▶ Code changes, Deployment Descriptor updates, Module additions/removals, changes to WebSphere Process Server artifacts (business processes, human tasks, business rules, business state machines, enabling monitoring)
 - ▶ Utilizes WebSphere Application Server's Fine Grain Application Update support for J2EE artifacts
 - ▶ Utilizes WebSphere Process Server update/install support for WPS artifacts
- Manual
 - ▶ Application can be removed and added again through Add/Remove Projects to pick up changes
 - ▶ Restart Project will publish new changes
 - ▶ Restart Server when restart is invoked workspace
- Automatic publish
 - ▶ Enabled through setting Server document

▼ Publishing from

Modify the publishing settings.

Run server with resources within the workspace

Run server with resources on Server

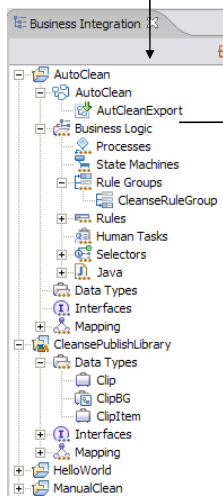
Enable automatic publishing

Publishing interval (in minutes): 1

As previously mentioned, when application binaries are run from the workspace, certain changes will be propagated automatically when updates are made to the application. If the binaries are run from the server, you will need to republish whenever updates to the application are made. Update publishing can be configured to occur manually or automatically. WebSphere fine grain application update is used to publish changes, so only the parts or modules of the application that have been modified will be updated. If you are making frequent changes to an application, it can be beneficial to disable automatic publishing. In this case, you would have to either remove and reinstall the application using add/remove projects, restart the project, or restart the server in order for the changes to be propagated. Automatic publish will periodically examine modifications that you have made to the application and publish them.

Automatic Publish – Run on Server

Add/Change/Remove artifact or module



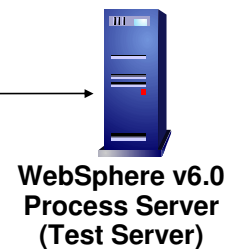
- Changes to applications are captured in preparation for publish

Builder runs and starts AppInstaller

AppInstaller determines type of change (“delta”)



“Delta” published to server



**Does not apply to resource running out of workspace*

This slide illustrates the manner in which changes to an application are propagated to the test server. As part of the build process, the changes which are made to the projects or applications will be noted and tracked. At publish times, the changes will be published to the server and WebSphere fine-grained update will apply the updates. Even when an artifact is deleted from a project or application a deletion package is sent to the server and read to delete the appropriate artifact.

Testing Applications Notes and Limitations

- Add/remove projects dialog will report application as installed even if application fails to install
- If WebSphere Integration Developer install path is too long or profile install path is too long, combined with long hostnames or long application path lengths, application installations/updates may fail
 - ▶ Hard to detect
- Console view is refreshed with SystemOut.log every ~5 seconds
 - ▶ Log file is not verbose
- All messages may not appear immediately in Console view
- All messages may not be printed to the Console view as the server is shutdown
- Console view may show out of date information
 - ▶ Old messages in SystemOut.log and SystemErr.log may be shown in view when contents are combined for display



By design, if there is a problem with installing an application, the add/remove projects dialog will still report that the application has installed.

Long install paths can cause errors, such as application install failures or update failures. Because of Windows operating system limitations, these errors may be reported as path not found, or class not found, making the real cause difficult to determine.

The console view is refreshed from the SystemOut and SystemError log files every few seconds, but it is not a verbose feed of the system logs. At intervals, the system log messages are retrieved from the server through the file transfer application. The plug-in does some analysis and sorts the messages; sometimes that changes the sort order of the messages. Be aware that the console messages are processed this way, and that a message may show up in the console several seconds after the event actually occurs.

Agenda

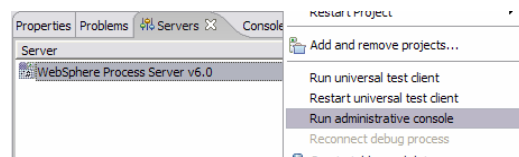
- Test Environment Architecture
- Installing Test Environment
- Preparing Test Environment
- Testing Applications
- **Working with Server Configuration**
- Summary and References



This section covers working with server configurations.

Configuring Server Resources

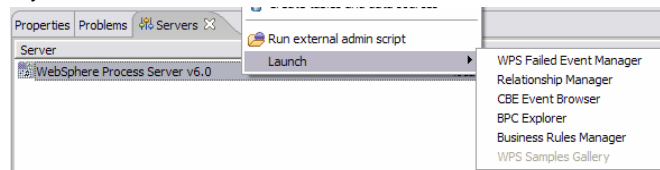
- The Administrative console is the primary interface for configuring WebSphere Process Server V6.0 test servers (local and remote)
 - ▶ Common interface for all servers
- Server Configuration editor replaced with Server editor
 - ▶ Server editor only contains details which points WebSphere Integration Developer
- Server must be started
 - ▶ Administrative Console does not need to be enabled



Any type of server resources or server settings, whether it is for the local test environment or a separate install, must be defined using the administrative console. This provides a consistent user interface as you move from test into production. The server configuration editor has been replaced by the server editor, which creates a server document that references a particular server. Due to the new architecture, the server must be started before you can administer it using the Administrative Console.

Additional Administrative Applications

- WPS Failed Event Manager
 - ▶ Viewing, deleting, and resubmitting failed events
 - ▶ Must log into the Administrative console and expand Integration Applications and select Failed Event Manager
- Relationship Manager
 - ▶ Viewing and rolling back Relationship instances and Role instances
 - ▶ Must log into the Administrative console and expand Integration Applications and select Relationship Manager
- CBE Event Browser
 - ▶ View CBE captured and saved by CEI
- BPC Explorer
 - ▶ Start business process instances and view human tasks
- Business Rules Manager
 - ▶ Only available if installed (non-default)
 - ▶ View and modify business rules installed on server



30

WebSphere Integration Developer Test Environment

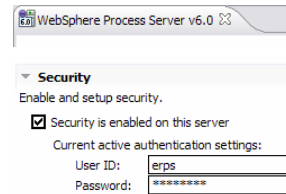
© 2005 IBM Corporation

Part of the test server support includes links to administrative applications for process server artifacts. Right-click on the server pointer document, and the Launch option displays a list of the administrative applications that you can run. The Failed Event Manager and the Relationship Manager are part of the Administrative Console; you will be prompted to log into the administrative console and then navigate to the appropriate application listed on the navigation bar under Integration applications.

The CBE Event Browser and the BPC Explorer can be found in the Administrative Console, but when launched from the Launch menu option, these will display in a window in the workspace. The Business Rule manager will only be enabled if the Business Rule Manager has been installed. The Business Rule Manager is something that must be installed after profile creation.

Setting up Security

- Security for test environment servers must be enabled through the Administrative Console
- User ID and password fields are used to authenticate to a running server to find status, publish applications, stop or restart servers
 - ▶ User ID and password are not used for execution of server
- SOAP setting should be used
- Console view will not display SystemOut and SystemError output until FileTransferSecure Application is installed
 - ▶ <http://www-1.ibm.com/support/docview.wss?uid=swg21198280>



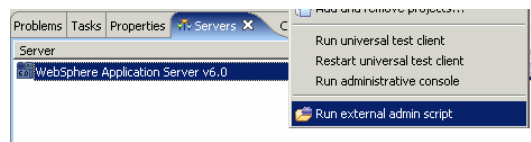
The server editor contains a section called security where you can enter the credentials needed to connect to your application server. You cannot actually enable security here (this must be done using the administrative console), you can use the “Enable Security” check box to indicate that security is enabled on the server and provide the credentials. These credentials are not used by the server to run; rather they are used by WebSphere Integration Developer to authenticate to the application server. Entering incorrect credentials here will not prevent the server from starting, but will cause problems when interacting with that server from WebSphere Integration Developer.

Rather than using RMI as a connector, use SOAP when security is turned on for the server.

The FileTransfer application that reads the logs from the server for the console view does not work with security enabled. To enable the console view to read from the server logs, follow the instructions at the URL listed at the bottom of the slide.

Running External Administrative Script

- Jacl or Jython scripts can be executed against a specific (started) test server
- Results outputted to the Console view
- No parameters can be specified to be passed to scripts



External administrative scripts written in JACL or Jython can now be run from within WebSphere Integration Developer. This will cause a wsadmin command to run, which connects to the SOAP port of the server that you had selected. The results will appear in the console view. One limitation to be aware of is that you cannot use a script that prompts for parameters.

Updating Test Environment

- The Update Installer for WebSphere Software is also known as the update installer program, the UpdateInstaller program, and the Update installation wizard
- WebSphere Application Server updates applied
- Update Installer available with Interim Fix
 - ▶ <http://www-1.ibm.com/support/docview.wss?rs=2307&context=SSQH9M&uid=swg24010506>



It is recommended that you update the test environment with an interim fix in order to resolve a number of issues. The update installer is included with this fix at the URL listed on the slide.

Known Problems and Limitations

- Backup “prof” directory after installation to lessen time to create new server configuration
- If problems with publishing remotely, try publishing locally with Resource run on Server setting
- Start server first before publishing applications
- Use SOAP setting for connecting to the server for more stability/reliability
- Hostname changes will cause the Console view to be blank
 - Can be corrected by entering old host name in HOST file
 - 127.0.0.1 <Old Hostname>
- If a Proxy is specified (Windows > Preferences > Internet > Proxy Settings), and the hostname of the system WID and the test server is installed on is not registered with the Proxy server, the Console view will be blank



This slide addresses some known limitations. The time it takes to create a new profile can be 8 to 15 minutes, depending on the system. It is simpler to create a backup of the profile by copying the directory; if necessary, you can have a clean profile by renaming the backed up copy to “prof”. This will require re-installing the applications, but is faster than creating a new profile.

If there are problems publishing remotely, try publishing locally. This will eliminate network issues and application errors as possible causes.

Start your test servers manually; this is faster than having WebSphere Integration Developer automatically start the servers when you publish an application.

Use the default SOAP connection rather than the RMI where possible.

If you change the hostname of the system, the Console view will no longer be able to retrieve the log entries. One workaround is to edit the HOST file to point the old host name to localhost as noted here.

However, if a proxy is specified, any lookups or resolutions will use the proxy server, even before using the local hostname file. If your system is not registered with that proxy server, then the Console view will not be able to locate the host name to retrieve the log file entries from. The proxy server will return a “host not found” message, and the Console will remain blank.

Agenda

- Test Environment Architecture
- Installing Test Environment
- Preparing Test Environment
- Testing Applications
- Working with Server Configuration
- **Summary and References**



And in summary...

Summary

- Consistent architecture for integrated (local) and remote servers
- Different publishing options for more automated deployment and publishing
- Server administration for integrated and remote servers through WebSphere Administrative Console



WebSphere Integration Developer V6.0 uses a consistent architecture that supports a variety of different test environment servers installed both locally and remotely. This presentation explained the different publishing options and the automation available for publishing. Finally, test environment server administration is done using the Administrative Console for local or remote servers.

Trademarks, Copyrights, and Disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

IBM	CICS	IMS	MQSeries	Tivoli
IBM (logo)	Cloudscape	Informix	OS/390	WebSphere
e (logo) business	DB2	iSeries	OS/400	xSeries
AIX	DB2 Universal Database	Lotus	pSeries	zSeries

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

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

Intel, ActionMedia, LANDesk, MMX, Pentium and ProShare are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a registered trademark of Linus Torvalds.

Other company, product and 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 and/or changes in the product(s) and/or program(s) 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.

Information is provided "AS IS" without warranty of any kind. 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 (e.g., 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 2005. 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.

