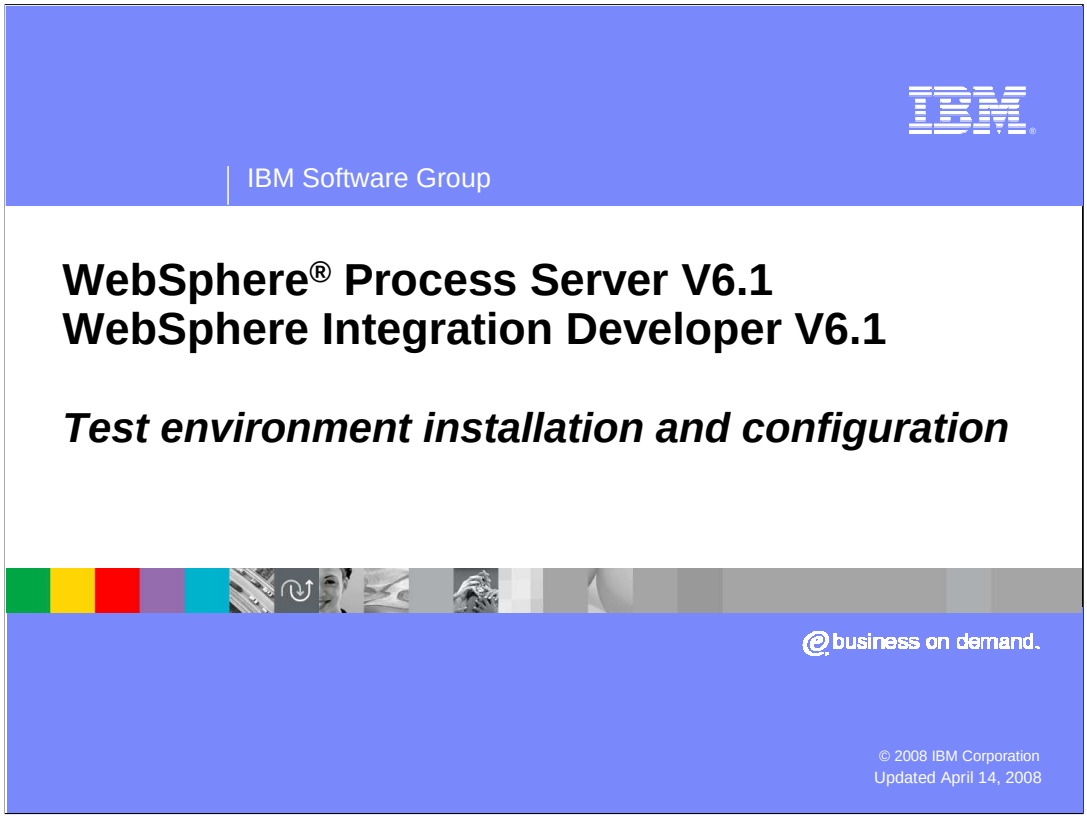


This presentation will discuss the details of the test environment included with IBM WebSphere Integration Developer V6.1 and support for testing applications on WebSphere Process Server V6.1.

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.

Section

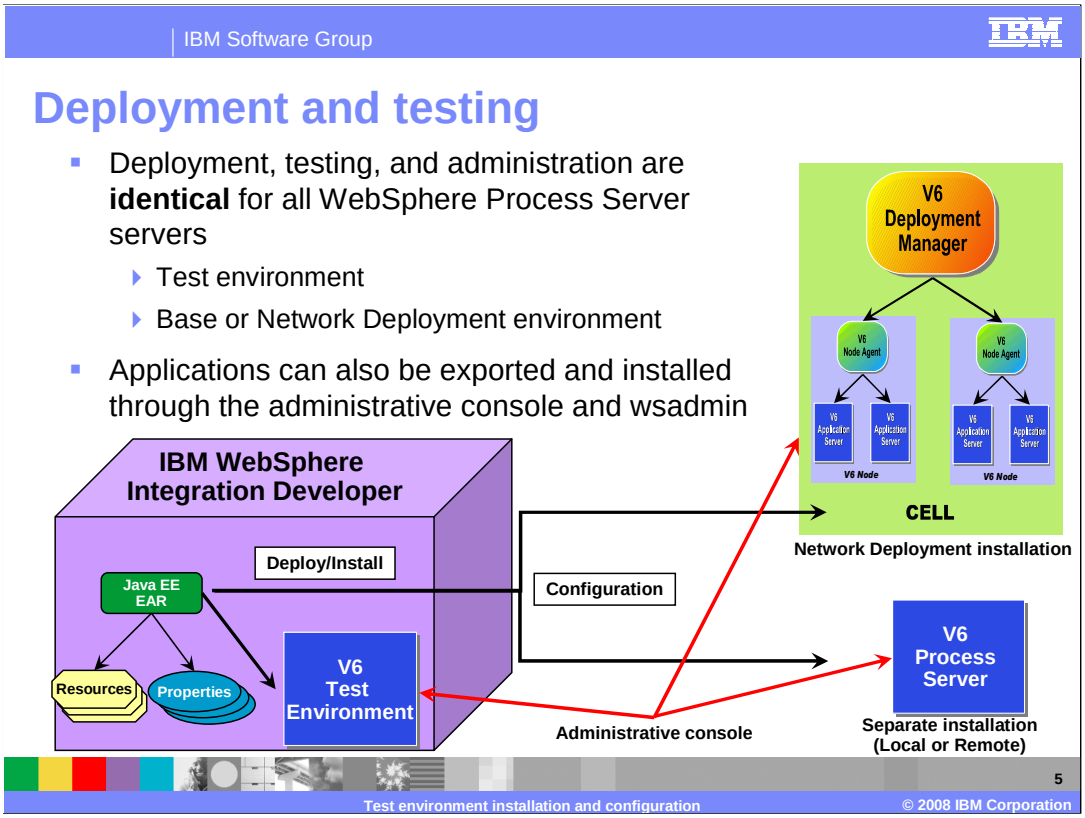
Test environment architecture

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
- Architecture to support deploying and testing on WebSphere Process Server and WebSphere Process Server Network Deployment
- Test environment configuration through administrative console for WebSphere Process Server

The test environment features a consistent deployment and administrative model for any server you might use. They are 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 the server configurations in the tools; the test environment is configured through the administrative console.



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, or a server in a cell are all treated the same way for deployment and administration. For deployment, IBM WebSphere Integration Developer allows 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 can also be exported from 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 a Java™ 2 Enterprise Edition EAR file before it is published.

Section

Installing the test environment

This section covers installing the test environment.

Test environment installation

- WebSphere Process Server test environment install option in WebSphere Integration Developer
  - Installed silently
  - Response files at disk6\external\WAS\
  - Business process choreography, CEI, application scheduler
  - J2C authentication alias set up with **admin/admin** for username/password

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 Disk6 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.

Test environment installation

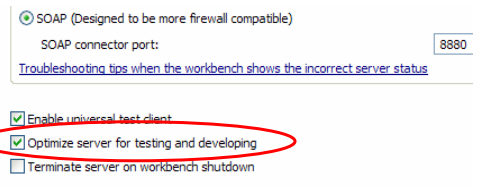
- Installation location
  - Runtime binaries located at ${WID Install Dir}\runtimes\bi_v61
    - Server Stub directory ${WID Install Dir}\runtimes\bi_v6_stub is not part of test environment install
  - Default configuration profile located at {$WID Install Dir}\pf
    - Profile named wps
- Install results located at ${WID Install Location}\runtimes\bi_v61\logs
  - Log.txt

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 pf directory. The profile name need not match the directory name – for the WebSphere Integration Developer, the profile name is “wps”. The reason for moving the pf directory up a few levels is because there are Windows® operating system imposed limits on path length. If the pf directory was under the runtimes\bi_v6 directory, then there can 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 consolidated with the WebSphere Integration Developer installation logs.

Test environment differences

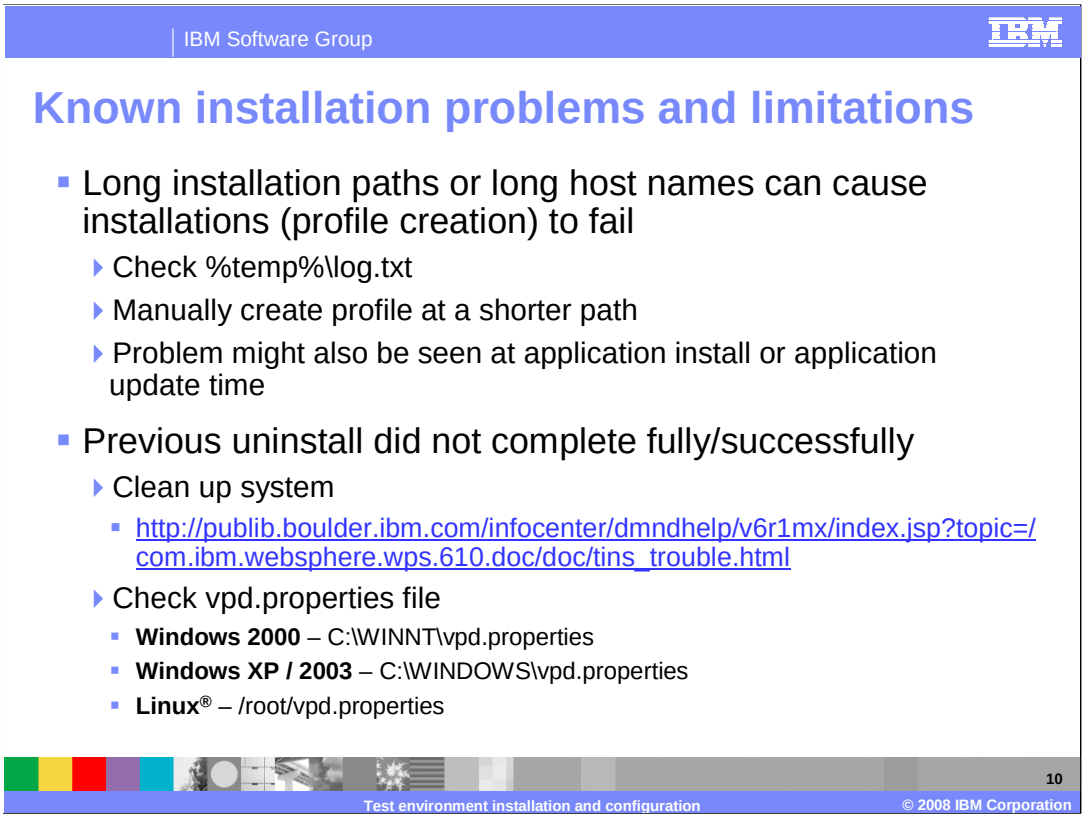
- Small differences between Test environment installation and stand-alone WebSphere Process Server installation
- Development mode on by default
  - Application servers > server1
- Transaction set to not time out (value 0)
  - Application servers > server1 > Container Services > Transaction Service
- Integrated test client support added to test environment



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 lifetime timeout value is set to zero, meaning no time out. This is useful when debugging and stepping through code. The typical 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 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.

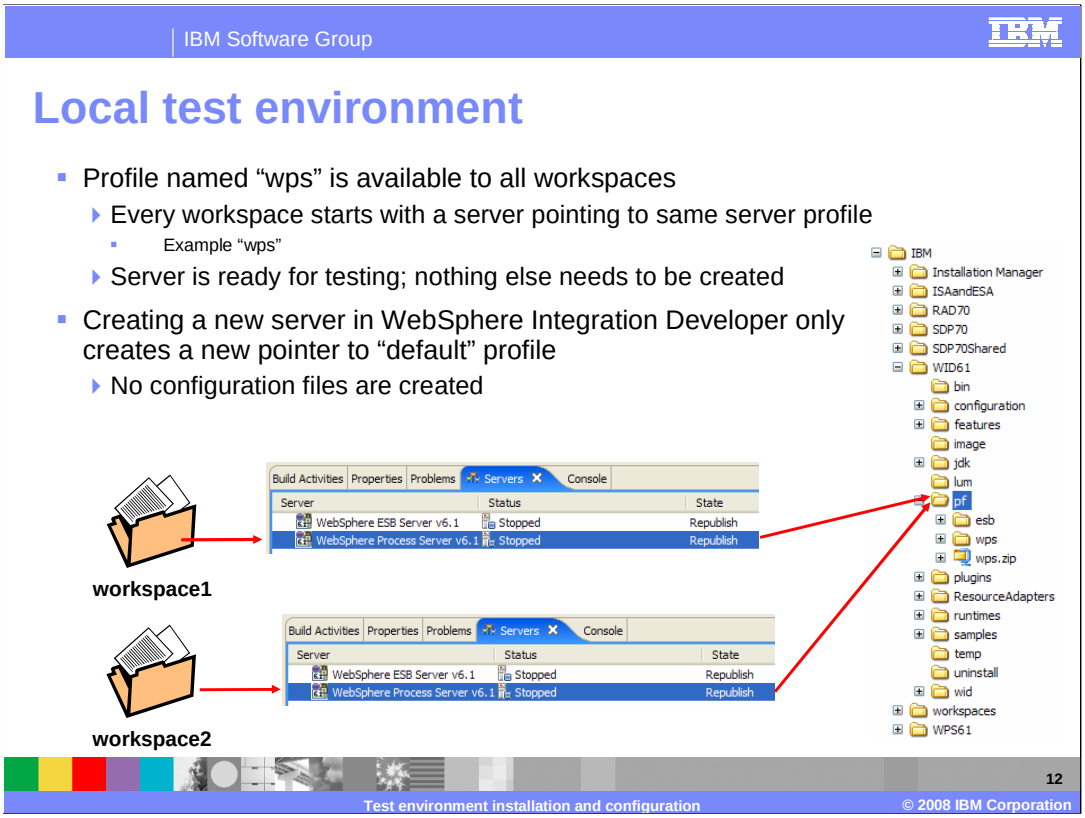
Long installation paths can cause the installer to fail; if so, check the log.txt file in the temp directory. You can also have path length problems later on when applications are installed or updated. Long path lengths do not always show up directly as errors; what you will see is an error message about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

Section

Preparing the test environment

This section covers preparing the newly installed test environment.

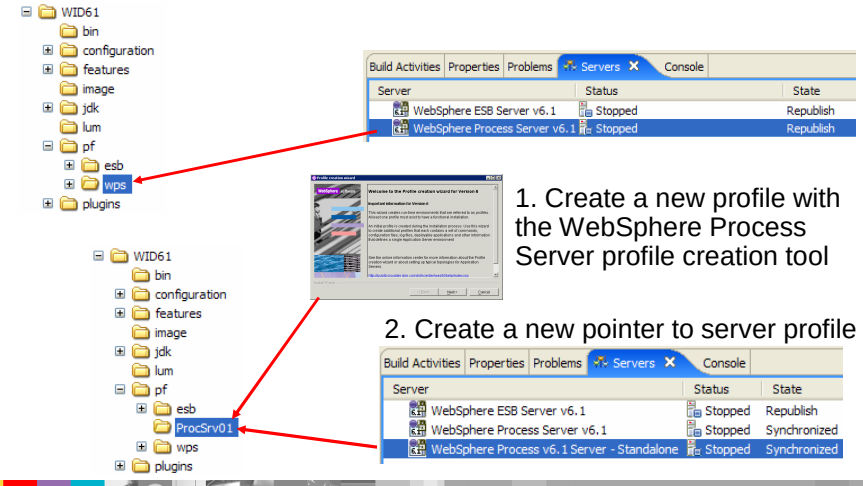


In the local test environment that is installed as part of WebSphere Integration Developer there is a default “wps” profile. Each workspace that you start will have a pointer to this profile. The profile is independent of the workspace, and all you have in the workspace is essentially a pointer to the profile. This means you might 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_v61\bin\ProfileManagement\bin\PMT.exe**



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 the local test environment

- WebSphere Integration Developer reads local test environment information from WebSphere 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_v61) or separate installation
  - profileRegistry.xml points to available profiles
    - ${WID_INSTALL_DIR}\runtimes\bi_v61\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_v61\bin

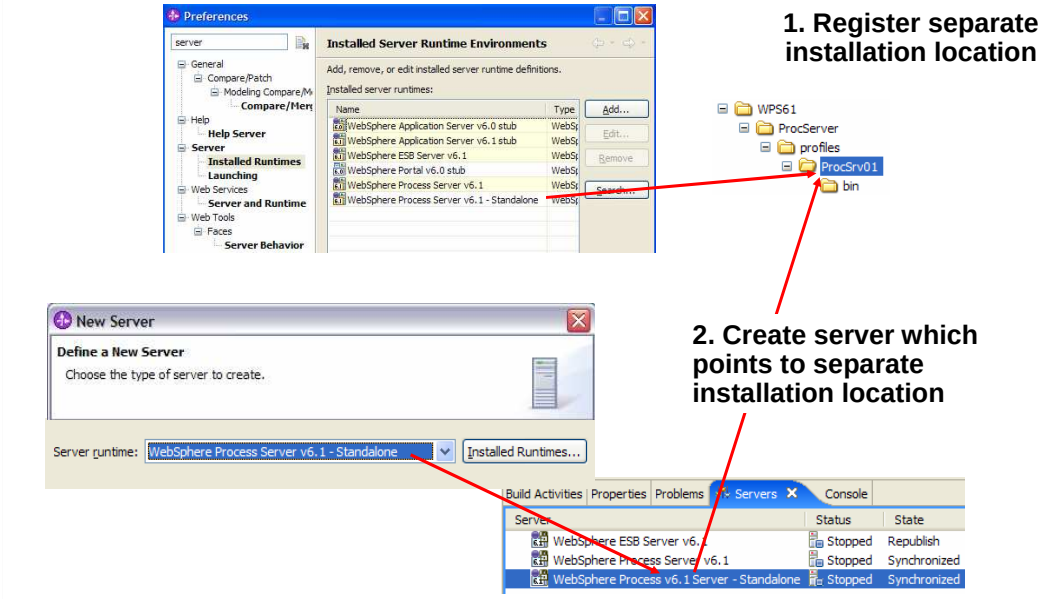
The workspace no longer contains the test environment configuration information. Instead, pointers are created to the test environment, whether it be the instance installed with WebSphere Integration Developer or an external installation of WebSphere Process Server. The profiles available for use as test environments are stored within the profileRegistry.xml file. This 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 deleting the profile directory. Manual editing can cause unpredictable results due to broken links and dependencies.

Local test environment: Separate installation

- Separate installations of WebSphere Process Server V6.1on the same system as WebSphere Integration Developer can be used as test environments
- Installed server runtime environment must be configured with location of separate installation
  - Windows > Preferences > Server > Installed Runtimes
- New server in WebSphere Integration Developer must point to correct profile of separate installation
  - 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 configure WebSphere Integration Developer to use 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 want to use as your test environment.

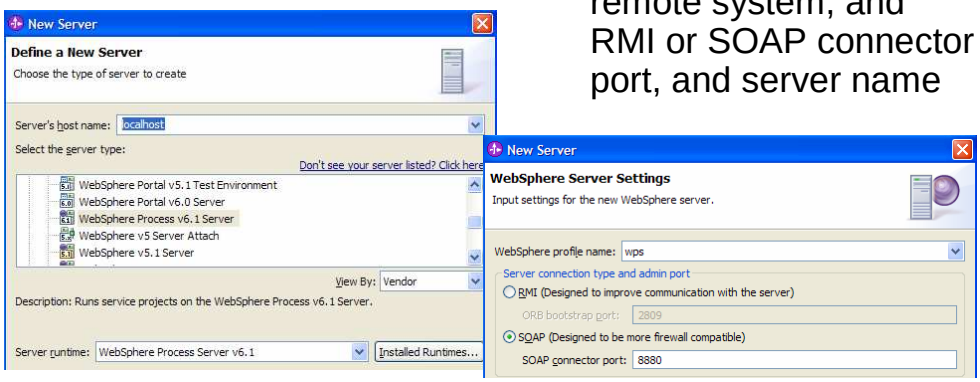
New test environment: separate installation



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 can also be configured as test environments
  - All platforms

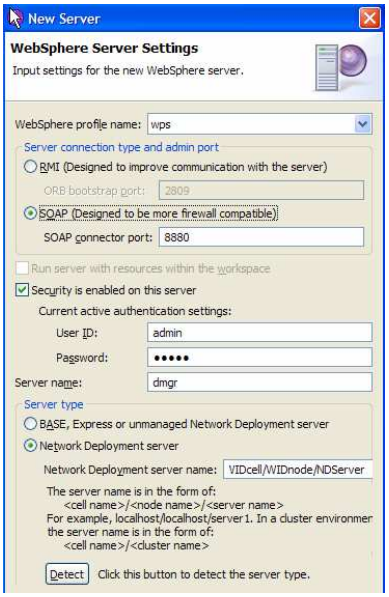


- Specify host name of remote system, and RMI or SOAP connector port, and server name

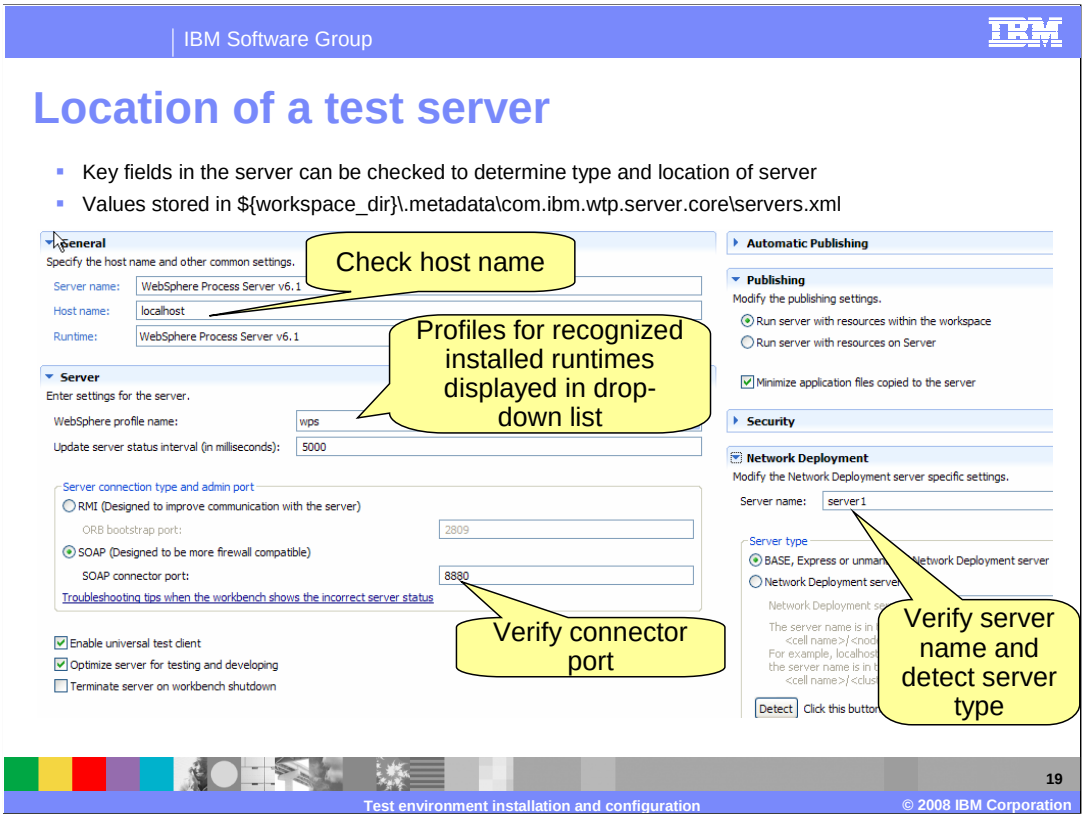
Remote instances of WebSphere Process Server are supported for use as test environments. The remote test environment is very simple to configure. When creating a new server document or pointer document, specify something other than localhost for the Host name and the SOAP connector port for the particular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

Preparing to test on network deployment

- Servers which are part of a Cell (Network Deployment 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



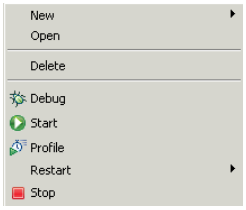
Testing with a server that is part of a Network Deployment cell environment is supported in V6.0 as well. For this type of configuration, you will again specify something other than localhost for the Host name and the SOAP port for the deployment manager node. The SOAP connector port for a deployment 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.



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 servers
  - Restart in different modes (Normal, Debug, and Profile)
- Stop
  - Stops running server



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 installation, 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 create a new process, which causes the restart. This eliminates the need to use the remote agent controller.

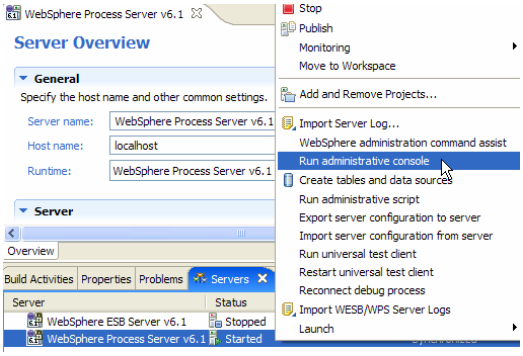
Section

Working with server configuration

This section covers working with server configurations.

Configuring server resources

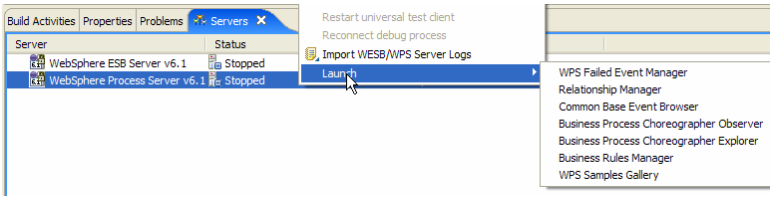
- The administrative console is the primary interface for configuring WebSphere Process Server test servers (local and remote)
  - Common interface for all servers
- Server configuration editor replaced with server editor
  - Server editor only contains details which points WebSphere Integration Developer
- Server must be started to use administration console
  - 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. The server must be started before you can administer it using the administrative console.

Additional administrative applications

- WebSphere Process Server 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
- Common base event browser
  - View common base events captured and saved by the common event infrastructure
- Business process choreographer observer
  - Observe the states of running processes and human tasks
- Business process choreographer explorer
  - Administer business process and human tasks objects such as process instances, failed activities, and work assignments
- Business rules manager
  - Only available if installed (non-default)
  - View and modify business rules with templates installed on server

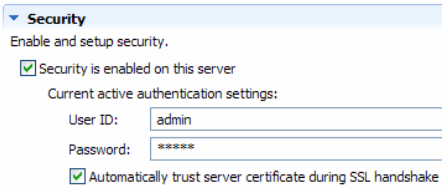


Part of the test server support includes links to administrative applications for process server artifacts. Right-click on the server pointer document, and the Launch option displays a list of the administrative applications that you can run. The failed event manager and the relationship manager are part of the administrative console. You are prompted to log into the administrative console and then navigate to the appropriate application listed on the navigation bar under Integration applications.

The common base event browser, the business process choreographer observer, and the business process choreographer 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 it 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 running the server
- SOAP setting should be used

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.

.

Known problems and limitations

- Backup “pf” 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
- Use SOAP setting for connecting to the server for more stability/reliability
- Host name changes will cause the console view to be blank
  - Can be corrected by entering old host name in HOST file
    - Example: 127.0.0.1 <Old Host name>
- Console view is blank
  - A proxy is specified (Windows > Preferences > Internet > Proxy Settings)
  - And host name of the system is not registered with the proxy server

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 “pf”. 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 host name 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 host name file. If your system is not registered with that proxy server, then the Console view will not 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.

Section

Summary

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 the administrative console

WebSphere Integration Developer V6.1 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.

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_WBIV61_WID_TestEnvironment.ppt

This module is also available in PDF format at: ../WBIV61_WID_TestEnvironment.pdf

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

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 WebSphere

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

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

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

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.

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 (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.

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.