



IBM Software Group

IBM® WebSphere® Application Server V6

Web Server Definition and Plug-in



@business on demand.

© 2005 IBM Corporation
Updated April 28, 2005

This presentation will focus on support for Web Servers in WebSphere Application Server V6.

Goals

- Understand Managed and Unmanaged nodes
- Introduce Web Server definitions in WebSphere Application Server topologies
- Understand IBM HTTP Server (IHS) V6 Administration
- Understand Plug-in file generation and propagation



The goal of this presentation is to introduce the core concepts of Web Servers in WebSphere Application Server V6. In particular this presentation will cover the new concept of managed and unmanaged nodes within a WebSphere topology. There will also be an explanation of Web Server administrative tasks.

Agenda

- Web Server definition overview
- Web Server definition for Stand-alone Application Server
- Web Server Definition in a V6 Network Deployment cell
 - Deployment Manager Administrative Console UI to manage Web Servers and the plug-in files
- Create and manage Web Server definitions
- IBM HTTP Server V6 management from WebSphere V6
- Plug-in file generation and propagation
- z/OS Current Points and Work Arounds



The presentation will begin with an explanation of how to create a Web Server definition in an ND cell. V6 offers the option to create a managed or an unmanaged node for the purposes of managing the Web Server from within a WebSphere topology. The WebSphere administrative console provides the capability to manage both the Web Servers and their plug-in files. Next, Web Server creation in a Stand-Alone or Deployment Manager environment will be discussed as well as how to manage an IBM HTTP Server through the WebSphere Application Server administrative console. Finally, Plug-in file generation features and propagation in WebSphere Application Server V6 will be discussed.

Section

Web Server Definition - Overview



This section will cover the basics of Web Server definitions.

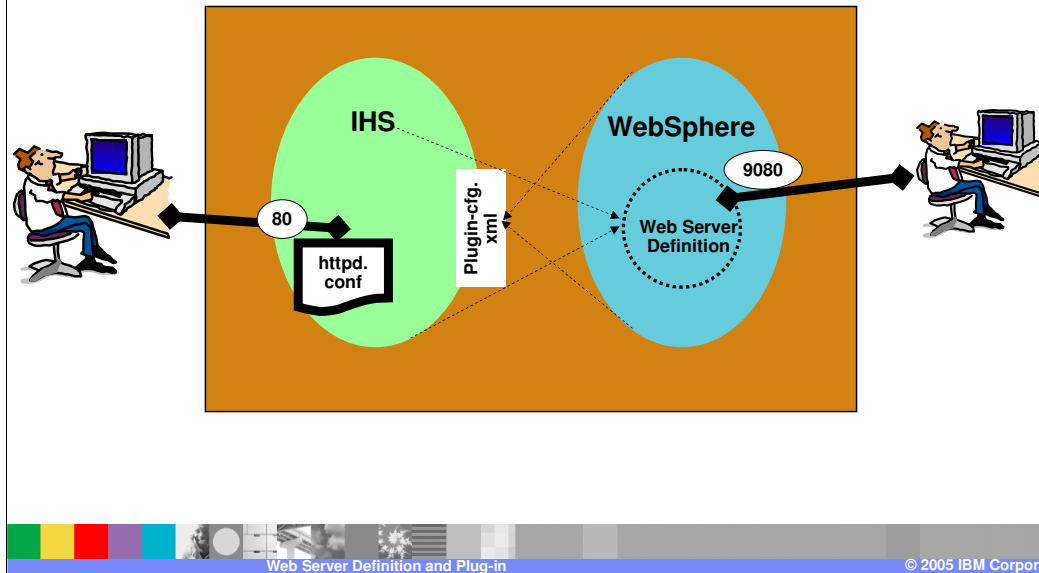
Web Server Definition: Overview

- Web servers can now be defined in a WebSphere Application Server topology on a Managed or Unmanaged Node
 - ▶ Managed Node has a Node Agent through which the node is managed by the Deployment Manager (DMgr)
 - ▶ Unmanaged Node has no Node Agent and is not managed by DMgr
- Applications can be associated with one or more defined Web servers
 - ▶ This allows generation of custom plug-in configuration files for a specific Web server
- A Stand-alone Application Server topology is limited to just one unmanaged Web server



The ability to define a Web server configuration within a node is new in version 6. In the case of an Apache based IBM HTTP Server, the WebSphere administrative console can be used to stop and start the Web server and transparently update the plug-in configuration file on a remote machine where the Apache based Web server is installed. WebSphere can also selectively target applications to specific Web servers, so that only specific Web servers can route requests to that application. Web Servers can be defined on managed or unmanaged nodes. The concept of a managed node means that a node agent is running on the Web Server system. With an unmanaged node there is no node agent, which can be useful if you want to place the Web Server in a DMZ. IBM HTTP Server on z/OS platforms is based on Domino Go and not Apache, so these capabilities are restricted to generating plug-in files for these Web Servers.

Web Server Definition



Web Server Definition and Plug-in

© 2005 IBM Corporation

6

This graphic presents a stylized view of Web servers and their definition within the WebSphere repository.

The plugin-cfg.xml file contains directives that control the forwarding of requests between the HTTP Server and WebSphere Application Server. The plug-in file can be generated from the Administrative Console (using port 9080 in the picture) or through a special script. The plugin-cfg.xml file must be manually defined in the Web server httpd.conf file as a plug in. IBM HTTP Server (IHS) must be installed and configured as usual before the plugin-cfg.xml file definition is added to the httpd.conf file. The Web server is managed by an administrator (using port 80 as shown here).

The new administrative function provides an alternative way to administrate the Web server. It can still be controlled as before by using the Web server administrative interface or manually editing the httpd.conf, but it can now also be controlled from the WebSphere administrative console. Both the httpd.conf and the plugin-cfg.xml files can be modified from the WebSphere Administrative Console.

This picture can now be generalized beyond the stand-alone case. For example, the Web server and the application server could be on separate hosts. The web server could even be on a distributed host.

You can even have multiple definitions corresponding to multiple Web servers.
 WASv601_zOS_WebServer.ppt Page 6 of 47

Web Server Support: At a Glance

WebSphere Topology Applicability	Web Server Support	Requirement	Web Server Administration Capability
All packages - Stand-alone Application Server (ND or Express) or ND Cell	Un-managed Web Server Node This is same as support in WebSphere v5.x	None	None
Network Deployment (ND) Cell only	Managed Web Server Node	Requires Node Agent running on the Web Server machine	Auto propagation of plug-in configuration file to any remote Web Server machine
	IHS as a special case of Unmanaged Node	Requires Apache Based IHS AdminService to be running	If using Apache based IHS, then you can start/stop IHS



This slide details the support for managed and unmanaged nodes based on the version of WebSphere you use. The key point to notice here is that in a stand-alone server environment, there can only be a Web Server definition on an unmanaged node.

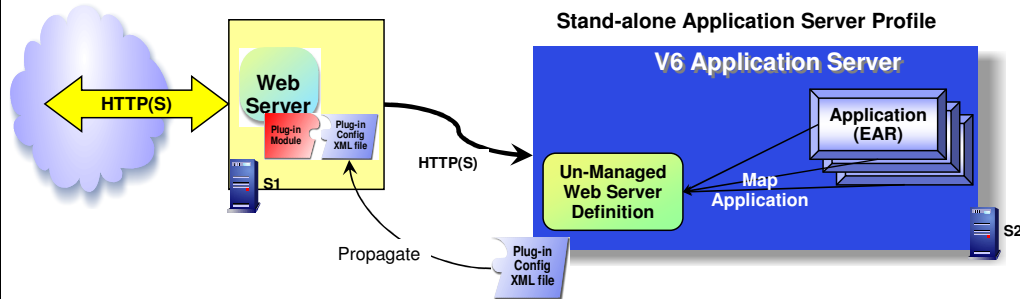
Section

Web Server Definition for a Stand-alone Application Server



This section will show the support for a Web Server definition in a Stand-alone application server topology.

Web Server Definition: Stand-alone Server



- For local Web server plug-in install, a Web server definition is created during plug-in install, for default Stand-alone Application profiles
- For other Web Server plug-in install cases and other profiles, a command script is generated that can then be used to create the Web server Definition
- Web server definition is created under an unmanaged node – It cannot be defined under the node in which the application server is defined
- Generated plug-in configuration file can be automatically propagated to a remote Apache Based IBM HTTP Server V6.0 using Web server Administrative interface.
 - For all other Web Server types, plugin-cfg.xml file will need to be copied manually to remote machine
- Web Server cannot be managed from the WebSphere Administrative Console

The Stand-alone application server environment only supports the notion of a Web server in an unmanaged node, which is shown in this diagram. When the plug-in is installed on the system with the application server, a Web server definition is created within the configuration. The plug-in configuration file must be propagated to the system where the Web Server is installed. Propagation is supported for the Apache Based version of IBM HTTP Server V6.0 using IHS administration server. WebSphere V6 has the capability to map applications to specific Web Servers. However, in this case since there can be only a single Web server, all applications will be mapped to the same Web Server.

Section

Web Server Definition in a Network Deployment Cell



This section will detail a Web server definition in a Network Deployment setting.

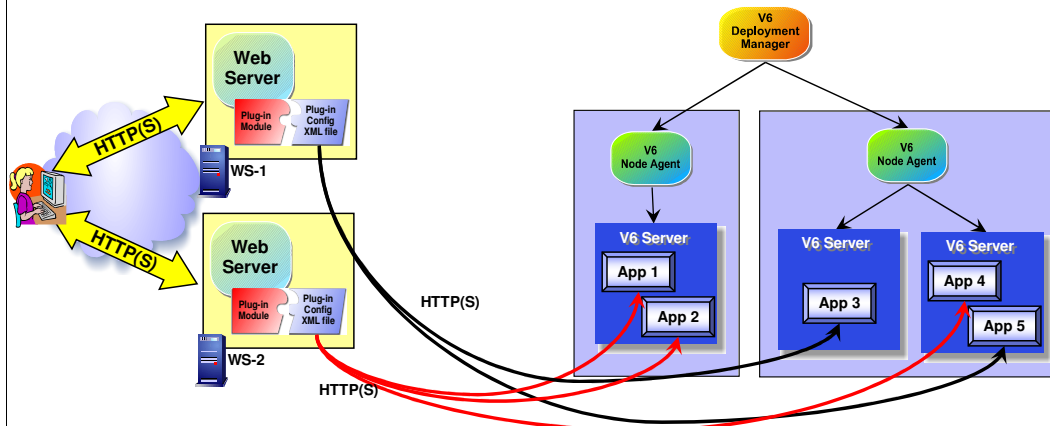
Web Server Definition in a ND Cell

- Web servers are defined as Web Server nodes in Cell topology under Managed or Unmanaged Nodes
- Following administrative tasks are supported
 - ▶ Create a new Web server definition templates
 - ▶ Create new Web server definitions from pre-defined templates for different supported Web servers
 - ▶ Manage and administer the managed apache based IBM HTTP Server Web servers
 - Start, Stop, Delete, Modify Web server configuration
 - ▶ Customize Plug-in configuration (plugin-cfg.xml) file for each Web Server
 - Map applications to a Web server
 - Configure Plug-in configuration properties (like Error level, etc.)
 - ▶ Propagate Plug-in configuration files



In a Network Deployment environment, Web Servers can either be defined as managed or unmanaged nodes. As a managed node, there will be a node agent present that will allow the deployment manager to manage the Apache based Web server. As an unmanaged node or a non Apache-based Web server, there is no node agent so the deployment manager is more limited in managing the Web server. Administrative tasks allow you to create new Web Server definitions from templates that have been defined. A new Web server definition template can be created from an existing Web server definition. The administrative console also provides enhanced functionality to customize the plug-in configuration files for a Web server. The plug-in can then be automatically propagated to Apache-Based managed Web servers or manually copied to unmanaged or Domino Go Web servers.

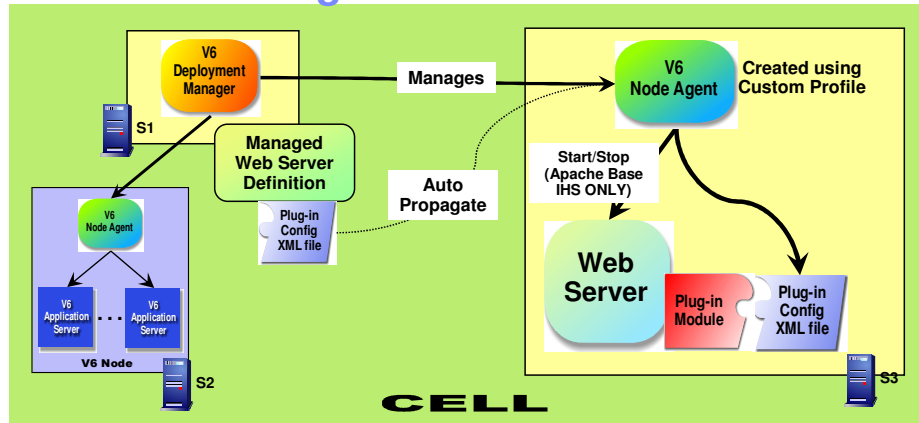
Web Server Topology: Application Centric



- Applications App-3, App-5 are mapped to Web server WS-1
- Applications App-1, App-2, App-4 are mapped to Web server WS-2
- Custom plug-in files will automatically be generated for Web servers to send requests for only their mapped applications

This slide shows how applications can be mapped to specific Web servers. These Web servers will then be responsible for handling the requests for the application that are mapped to that Web server only. These mappings result in the Deployment Manager creating the custom plug-in files. The information contained in the plug-in can also contain other customizations, such as caching and balancing features.

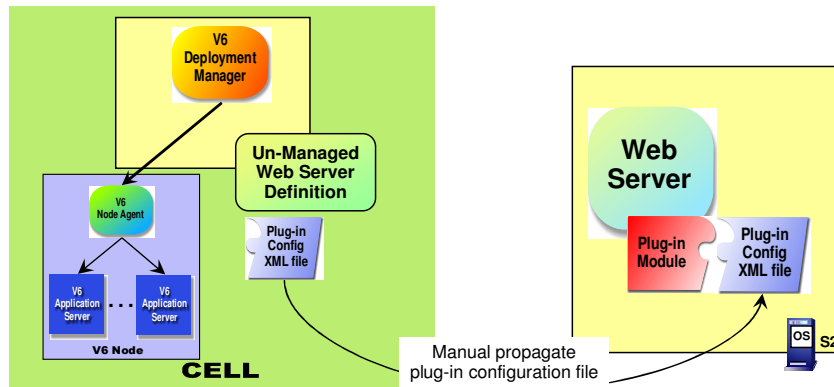
Web Server: Managed Node in a ND Cell



- Manage Web server (defined on a Managed node) from WebSphere DMgr
- Requires a WebSphere Node to be created on the Web server machine. Node Agent receives commands from DMgr to administrate the Web Server
- Provides ability to start and stop the Web Server and automatically push the plug-in configuration file to the Web Server
- Common for web servers installed behind a firewall where a WebSphere Node can be installed

This slide shows an example of a Web server in a managed node. A node agent is present on the system where the Web server is installed. The Apache-based Web server is managed by the WebSphere Deployment Manager through the node agent. The node agent provides the capability to start and stop the Web Server as well as automatically propagate the plug-in configuration file to the Web Server. This scenario is most common for a Web server installed behind a firewall where a WebSphere Node can be installed without any security concerns.

Web Server: Unmanaged Node in a ND Cell



- Web server not managed by WebSphere (defined on a unmanaged node)
- Manually propagate the plug-in configuration file from the DMgr system to the Web server system. The Apache based IBM HTTP Server allows for more management in an unmanaged node, which will be explained later in this presentation.
- Allows system administrator to create custom plug-in files for specific Web server
- Common for Web servers installed outside firewall or in DMZ, where no Node Agent is installed

In this example the Web server is defined in an unmanaged node and registered as an Unmanaged Node in WebSphere configuration. This allows a WebSphere System Administrator to create custom plug-in files for that Web server. This is covered in more detail in the plug-in presentation. When the plug-in is created it must be manually copied to the Web server. The deployment manager has no capability to direct the Web server. A Domino Go- based Web server falls into this category as well. The Apache-based IBM HTTP Server allows for more management of an unmanaged node, this will be explained later in this presentation.

Section

Creating and Managing Web Server Definitions and Mapping Applications to Web Servers



This section will demonstrate how to create and manage Web server definitions .

Creating Unmanaged Node for Web Server

The screenshot displays the IBM WebSphere Administration Console interface. On the left is a navigation tree with 'Nodes' selected. The main area shows the 'Add Node' dialog with 'Unmanaged node' selected. A yellow callout bubble points to this selection with the text 'Create Unmanaged Node'. Below the dialog, a yellow text box states: 'After you have created an unmanaged node, any new Web Server defined for that node is by default an unmanaged Web server'. To the right, the 'Nodes > New' configuration panel is shown, with 'Platform Type' set to 'Windows'. A yellow callout bubble points to this dropdown with the text 'Not Applicable on z/OS'. The bottom of the console shows 'Web Server Definition and Plug-in' and '© 2005 IBM Corporation'.

After you have created an unmanaged node, any new Web Server defined for that node is by default an unmanaged Web server

Not Applicable on z/OS

From the system administration panel, a new node can be created. This node can be either managed or unmanaged. The panels here show how to create a unmanaged node. A managed Node is created either through this panel, when creating a Custom profile, or when federating a Stand-alone application server.

Creating Web Server Definition

- Web Server definitions created during Web Server Plug-in install (on distributed platforms)
 - ▶ For Local Web Server Plug-in install and if the profile is an Express Stand-alone Application Server profile, the installer will create the Web Server definition within the Application Server configuration
 - ▶ For all other Scenarios, installer creates a JACL script, called `configure<WebServerName>.jacl`, which can then be modified and run for a specific WebSphere installation.
- For Network Deployment Cell, Web Server definitions can be created using DMgr Administrative console
- Using wsadmin: “createWebServer” administrative task can be used to interactively create Web Server definition
 - ▶ `wsadmin $AdminTask createWebServer -interactive`



Web server definitions can be created during Plug-in install on distributed platforms, using the Administrative Console in a Network Deployment environment, or using the wsadmin command line tool. The installation on a Domino Go-based z/SO IHS is always manual. The plug-in install either creates the Web Server definition or a JACL script called `configureWebserverDefinition.jacl` that enables you to create the Web Server definition later. The JACL script is useful for remote Web Server plug-ins, where the installer may not be able to access the remote machine.

configureWebserverDefinition JACL script

- Plug-in installation wizard under the covers use configureWebserverDefinition JACL script to create Web Server definition.
- configureWebserverDefinition script can be found in the <WAS_INSTALL_ROOT>/bin directory
 - ▶ This can be copied to the WebSphere install machine
- ConfigureWebServerName.sh script requires all the parameter necessary to create Web Server definition
- configureWebserverDefinition script will
 - ▶ Create unmanaged node if it does not exist
 - ▶ Create Web Server definition under the specified node
 - ▶ Map all the applications to the newly created Web Server
- Please see the usage documented in the configureWebserverDefinition.jacl script for complete detail on the options



On distributed platforms, the plug-in installer wizard uses the configureWebserverDefinition JACL script to create Web Server definition in the background. This slide describes the details of the JACL script, which is saved in the Web Server plug-in install bin directory. There is a special script created in the os390 directory on the distributed platform that can be used to create a Web server definition on a WebSphere for z/OS version 6.0.1 deployment manager.

IBM Software Group IBM

Create/Manage New Web Server Definition

1 Servers

- Core groups
 - Application servers
 - Generic servers
 - JMS Servers
 - Web Servers**
 - Clusters
 - Core group bridge settings
 - Cluster topology

2 Web Servers

A list of installed Web servers.

Preferences

Generate Plugin Propagate Plugin **New** Delete Templates... Start Stop Terminate

Select Name Node Version Status

Total 0

3 Create a new Web server entry.

Step 1: Select a node

Step 2: Enter the properties for the new Web server

Step 3: Select a Web server template

Step 4: Confirm new web server

Select a node

Select a node that corresponds

Select node

Unmanaged Node 1
Unmanaged Node 2
 RMGNode1
 Unmanaged webServer

Show Me

Specify Web Server Node (Managed or Unmanaged)

4 Create a new Web server entry.

Step 1: Select a node

Step 2: Enter the properties for the new Web server

Step 3: Select a Web server template

Step 4: Confirm new web server

Enter the Web server properties

* Type
IHS

* Port

Installation path

* Service name

Use secure protocol

Supported Web Server types

IHS
 APACHE
 IIS
 SUNJAVASYSTEM
 DOMINO

Web Server Definition and Plug-in © 2005 IBM Corporation 20

The steps for creating a Web server definition using the Administrative console is shown on this page:

Panels (1) and (2) show how to start creating the Web Server definition. When a new Web server definition is created, it must be associated with an existing node.

In Panel (3), the managed or unmanaged node on which the Web server will be defined is selected. In the example shown here, there are two possible nodes, both a managed and an unmanaged node have already been created in this topology. After a node is selected, the properties for the new Web server are entered. In panel (4), details for the Web server are provided. Click on the Show-me for a demonstration of creating a Web Server definition and mapping applications to the Web Server.

Managing Defined Web Server

- Select Web Server to configure the Web Server and its plug-in configuration file

Web Servers

Web Servers > Web Servers

A Web server that provides HTTP and HTTPS support to application servers.

Configuration

General Properties

Web server name
Web Server 1

Type
IHS

Use a secure protocol

* Host name
localhost

* Port
80

* Installation path
C:/IHS

* Configuration file name
C:/IHS/conf.httpd.conf

* Service name
IBMHTTPServer6.0

Additional Properties

- Remote Web server management
- Process Definition
- Plug-in properties
- Custom properties
- Configuration File
- Log file

Not applicable on z/OS

Web Server host and port

Not applicable on z/OS

Web Server Configuration file name

Not applicable on z/OS

Edit Plug-in configuration properties

21

Web Server Definition and Plug-in © 2005 IBM Corporation

Once a Web server has been added, an administrator can access the plug-in for that Web server through the administrative console. From this section an administrator can configure the plug-in.

Mapping Applications to Web Server

- Select application to map the modules to Web servers
- Select the Application Server and one or more Web servers
- Can also be done using the following wsadmin script:
 - ▶ `configureWebserverDefinition`
`.jacl`

Additional Properties

- [Stateful session bean failover settings](#)
- [Session management](#)
- [Application profiles](#)
- [Libraries](#)
- [Target mappings](#)
- [Last participant support extension](#)
- [View Deployment Descriptor](#)
- [Provide JMS and EJB endpoint URL information](#)
- [Publish WSDL files](#)
- [Provide HTTP endpoint URL information](#)
- [Map virtual hosts for Web modules](#)
- [Map modules to servers](#)



Within the configuration for each application various application modules can be mapped to defined Web servers. An application can be mapped to more than one Web server as well.

Section

WebSphere Application Server V6 and IBM HTTP Server (IHS) V6



This section will detail the enhancements for the IBM HTTP Server in V6.

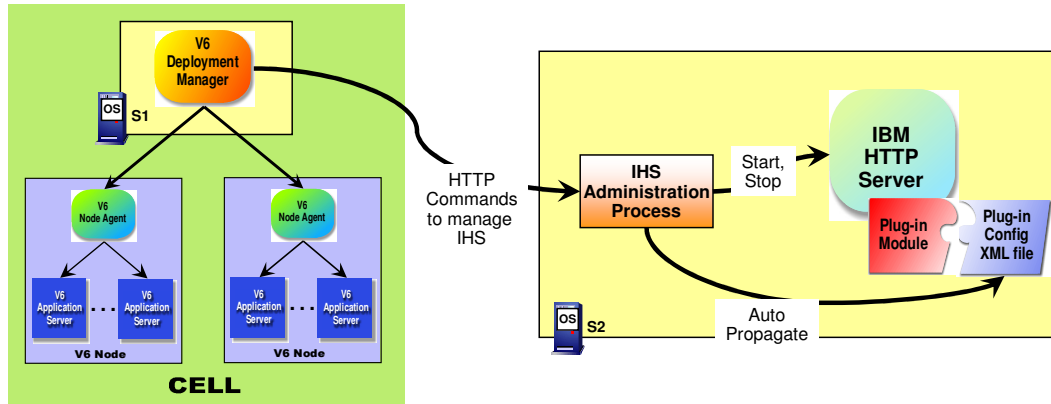
IHS Administration Overview

- Administrative functions for Apache based IBM HTTP Server (IHS) 1.3.X are handled by IHS Administrative Server (port 8008) through the IHS Administrative console
- Apache based IHS 2.0.X does not have any console administrative function.
 - ▶ Administration is done from the command line
- Apache based IHS V6 is bundled with WebSphere Application Server V6
 - ▶ Distributed platforms are based on Apache 2.0.47
 - ▶ Z/OS platforms are built on Domino Go
 - ▶ Administrative functionality is integrated into WebSphere Application Server V6 Administrative Console
 - ▶ Enhanced functionality with WebSphere Application Server



Earlier versions of the Apache based IBM HTTP Server were administrated through a separate administrative interface. Since version 2.0.x of the Apache based IBM HTTP Server this is no longer the case. Instead, administration is done from the command line. On Distributed platforms IHS V6 is bundled with WebSphere V6. More specific information on IHS can be found in the separate presentation for that product. On z/OS platforms, IHS is bundled with the z/OS operating system and can still be administrated through the Web based interface or through the WebSphere for z/OS v 6.0.1 administrative console.

IHS as Unmanaged (Remote) Node: Enhancements



- Provides administrative functions for IHS from WebSphere Application Server using the IHS Administrative process
- Provides ability to start and stop IHS and automatically propagate the plug-in configuration file to the IHS system and make configuration changes to httpd.conf
- Does not require Node Agent on the Web server machine

IBM HTTP Server based on Apache provides enhanced functionality for managing a Web server on an unmanaged node. This is accomplished through the use of a separate IHS Administrative process that runs on the same system as the Web server. The Deployment Manager communicates with the administrative process to manage the Web server. It might help to think of this separate administrative process as a smaller version of a node agent.

IHS Administration Server

- The IHS Administrative server runs as a separate instance of the IHS Web Server
- IHS Administrative Component for IHS 6.0 includes:
 - ▶ IHS Administration module (mod_ibm_admin.so or IBMModuleAdmin.dll).
 - ▶ IHS Administration configuration file (admin.conf)
 - Default port for the IHS Administration server is 8008.
- IHS Administrative authentication password file (htpasswd.admin)
 - ▶ Initially blank, which prohibits access to IHS Administration
 - ▶ Administrator updates IHS Administration password file using the htpasswd utility program
 - ▶ At a command prompt for /<IHS-Home>/bin enter the command:
 - `./htpasswd -m ../conf/admin.passwd <user_name>`
- To start and stop the administrative server
 - ▶ `<IHS-Home>/bin/adminctl start`
 - ▶ `<IHS-Home>/bin/adminctl stop`
 - ▶ Administer as a Windows Service



This slide details various commands and files that are needed to administrate the Apache based IBM HTTP Server. In particular, information is offered here on the separate administrative process as a reference.

Section

Web Server Plug-in Configuration file: Generation, Customization and Propagation



This section will detail the enhancements for IBM HTTP Server in V6.

Generating Plug-in with Administrative Console

The screenshot shows the 'Generate Plug-in' button highlighted in red. A yellow callout bubble points to it with the text 'Generate Plug-in'. Another yellow callout bubble points to the 'WebServer2' row in the table with the text 'Select the Web Server'.

Select	Name	Node	Version	Status
<input type="checkbox"/>	WebServer2	wsbeta026Node02	6.0.0.0	○
<input type="checkbox"/>	webserv1	wsbeta026Node01	6.0.0.0	✖

Total 2

- Plug-in configuration file can be generated using the Administrative Console after defining Web Server and mapping the Applications.
- Only Applications mapped to the Web Server will be added to the plug-in file (Application Centric)
- Plug-in files are generated automatically for any Application and Virtual host modifications. The default behavior can be changed by modifying plug-in property.
- Generated Plug-in files are always saved in the repository under Web Server defined directory.
 - ▶ config\cells\CellName\nodes\NodeName\servers\webServer1\plugin-cfg.xml

Once a Web server has been added, an administrator can generate a plug-in for it through the administrative console. Plug-in files generated for Web Servers will contain all the URIs for Applications mapped to the Web Server. By default, the plugin-cfg.xml files are generated automatically for any change to the Application or virtual host setting. Generated plug-in files are always saved in the master repository under the Web Server defined directory.

Generating Plug-in using command

- **GenPluginCfg.sh** command generates Web server plug-in configuration file, plugin-cfg.xml
- **Application centric** plug-in file generation
 - ▶ Used to generate plugin-cfg.xml file only for applications mapped to specific Web Server
 - ▶ Use `--webserver.name` option
 - Example: `GenPluginCfg.bat -webserver.name webServer1`
 - plug-in file for webServer1 and saved under webServer1 defined server directory in the master repository
- **Topology centric** plug-in file generation
 - ▶ Same as V5.x
 - ▶ Used to generate plug-in file for the entire cell, node or Application Server
 - ▶ Example: `GenPluginCfg -cell.name CellName -node.name appServerNode -server.name server1`
 - Above command will generate plug-in file for all the applications installed on the server1



The GenPluginCfg.sh command is used to regenerate the WebSphere Web server plug-in configuration file, plugin-cfg.xml. When the GenPluginCfg.sh command is issued with the `-webserver.name webserverName` option, the plug-in configuration file is created for the Web server. This setting in this generated configuration file are based on the list of applications that are deployed on the Web server. When this command is issued without the `-webserver.name webserverName` option, the plug-in configuration file is generated based on topology within the application server.

Customize Web Server Plug-in Properties

- Select Web Server to configure its plug-in configuration file

Web Servers > Web Servers > Plug-in properties

Configure Web server plug-in properties. The plug-in is used to pass HTTP requests from a Web server to WebSphere Application Servers.

Runtime Configuration

Plug-in properties

- * Plug-in installation location:
- * Plug-in configuration file name: [View](#)
- Automatically generate the plug-in configuration file
- Automatically propagate plugin configuration file
- Ignore DNS failures during webservice startup
- Refresh configuration interval: seconds

Plug-in logging:

- * Log file name:
- Log level:

Additional Properties

- Request and Response
- Caching
- Request Routing
- Custom Properties

30

Web Server Definition and Plug-in

© 2005 IBM Corporation

This is the administrative console screen used to configure the plug-in. Various additional properties can also be configured for the plug-in from the additional properties on the right. This should allow an administrator to perform most customizations without having to manually edit the plug-in configuration.

Propagation of the plug-in file

Generate Plug-in | **Propagate Plug-in** | New | Delete | Templates... | Start | Stop | Terminate

Select	Name	Node	Version	Status
<input type="checkbox"/>	WebServer2	wsbeta026Node02	6.0.0.0	⊙
<input type="checkbox"/>	webserv1	wsbeta026Node01	6.0.0.0	✖
Total 2				

- Plug-in file can now be propagated to the remote Web server system
- For a Web Server defined on a managed node, propagate action will invoke “nodeSync” operation. Node Agent will handle replicating the repository on the remote machine
- For a Web Server defined on an Unmanaged node , propagation is possible only with IBM HTTP Server
 - ▶ DMgr Administrative Console communicates with IHS Administrative Server configured on remote machine
- HTTP protocol is used for transferring the plug-in file

Once the plug-in configuration file has been created, it must be copied to the appropriate Web server. For an Apache-based Web server on a managed node this can be done by invoking the node sync operation through the WebSphere for z/OS v 6.0.1 administrative console. The node agent on the managed node will handle replicating the plug-in configuration file stored in the repository to the destination Web server on the remote machine. For a Web server defined on an unmanaged node, automatically copying the plug-in configuration file is only possible through the use of the separate administrative process of an IBM HTTP Server version 6 Web server. For other Web servers, including the Domino Go-based Web server for z/OS platforms, the plug-in configuration file must be manually copied into the appropriate directory on the remote Web server system.

Plug-in Propagation : At a Glance

Web Server Scenario	Requirement on remote Web Server machine	Destination after propagation
Z/OS (Domino Go) Remote Non-IHS Unmanaged Node	None	Files must be copied manually Recommended location: <Plug-in Install>/config/WebServerName/plugin-cfg.xml
Local Web Server Unmanaged	Configure Web Server directly where plug-in file are generated under master repository ¹	Propagation not required
Remote IHS Web Server Unmanaged Node	Requires IHS AdminService to be running, with write permission for destination directory	Files are propagated under Plug-in install directory on remote machine. <Plug-inInstall>/config/WebServerName/plugin-cfg.xml
Any Web Server under Managed Node	Node Agent should be running on the Managed Node	Files are replicated under node configuration directory. <WAS_Profile_Home>/config/cells/<CellName>/nodes/<NodeName>/Servers/<WebServerName>/plugin-cfg.xml

¹ Plug-in file for Web Server are generated under Web Server definition in the master repository

<WAS_Profile_Home>/config/cells/CellName/nodes/NodeName/Servers/WebServer1/plugin-cfg.xml

This slide details the plug-in automatic propagation and destination location on a remote system. Local Web Servers are configured to allow updates when the plug-ins are generated so no manual propagation is necessary. Local Web Servers on z/OS are configured by manually copying the plug-in configuration file to the appropriate location for the Web sever. If the web server has been configured to look for the plug-in configuration file in the WebSphere repository, no copying is required. If you are working with an Apache-based IHS Web Server configured as an unmanaged node , the files are propagated under the plug-in installation directory <Plug-in Install>/config/WebServerName/plugin-cfg.xml file. For Managed or Custom nodes the node sync operation takes care of replicating the configuration.

Section

z/OS Current Points and Work Arouds

This section will show how to create and manage Web server definitions .

Deployment Manager: New Configuration Panels

```
----- WebSphere Application Server for z/OS Customization -----
Option ==>

Define Variables to Configure Deployment Manager Node

Specify a number and press Enter to define the variables. You should
review all of the variables in each of the sections, even if you are
using all of the IBM-supplied defaults.
After you complete all sections, press PF3 to return to the main menu.

Completed?

1 - System Locations (directories, HLQs, etc)
2 - System Environment Customization
3 - Server Customization
4 - Web Server Configuration
5 - View Security Domain Configuration Panels
```

Added

There are new ISPF configuration panels to specify Web server definitions. The panel shown here is for Deployment Manager configuration. Similar selections appear in the other panels, such as for configuring **Stand-Alone** or **Federating a Node**. All the configuration panels for **Web Server Configuration** are optional. Their purpose is to allow you to create Web server definitions in the deployment manager using batch jobs. The same definitions can also be created using the Administrative Console.

Deployment Manager: Web Server Definition

```
----- WebSphere Application Server for z/OS Customization -----
Option ==>

Web Server Configuration (1 of 1)

Web Server Name: webserveronzDM
  The name used in defining the web server in the admin console.

Host..... : MVS223
  Hostname of the system on which the web server is located.

Port..... : 80
  Port on which the web server is listening.

Application Mapping ((A)ll, (N)one, (D)efault): A
  Determines whether you want to map all, none, or the Default
  Application to the web server.

Deployment manager security is enabled: N
  User ID..... : VGADMIN
```

This name is local to WebSphere

This automatically generates mapping definitions for the web server.



This slide shows the information required for a Web server definition in a Deployment Manager and again there are similar panels in **Stand-Alone** and **Federate a Node** panels. Some of the panels have different fields. The name here (webserveronzDM) is not related to any names or aliases inside the Web Server.

New JCL Generated

- BBO*CFGW
- Prevents federating a stand-alone Server
- Makes the Applications Invisible



The new panels each generate new JCL and instructions for running the jobs. However, after running the BBOWCFGW to create a Web server definition in a Stand-Alone server, you will not be able to federate the application server into a Network Deployment cell. The answer here is to not run this job against a stand-alone server that you plan to federate into an ND cell. On a Deployment Manager server, running the JCL can make the application apparently disappear. When you select **Enterprise Applications** in the administrative console, the list of applications will be empty. The easiest solution is to use either the wsadmin.sh tool or the Administrative Console to remove the Web server definition, after which the applications will re-appear on the list.

Working in the Administrative Console

- Unexpected results in Web Server Administration
 - ▶ Login under a different name
- Some Fields do not Make Sense
 - ▶ The Fields were only Designed for Distributed Platforms



If you are working with the Web Server Definition and not getting the expected results, try logging on under a different name. The system appears to keep some environment information related to Web Server Administration by user. The easiest work-around is to logoff the Administrative Console and log back in under a different name. Another option is to recycle the server. Several of the fields that are presented in the various panels are designed for distributed platforms and are not valid on z/OS platforms. These include installation path, platform type and the complete process definition panel, as well as others. You can see the mentioned fields in the following slide. This list is not comprehensive however, so there are others as well.

The screenshot displays the 'General Properties' configuration page for a web server in the IBM WebSphere Administration Console. The page is titled 'General Properties' and is part of the 'Web servers' configuration area. The 'Configuration' tab is active, showing various fields for the web server. A yellow callout points to the 'Installation path' field, which is set to '/usr/lpp/internet', with the text 'This field is not relevant in z/OS'. Another yellow callout points to the 'Configuration file name' field, which is set to '/etc/pcell2/internet/conf/httpd.conf', with the text 'Editing the httpd.conf file can cause an error'. The page includes a navigation menu on the left, a top header with 'IBM Software Group' and 'IBM' logos, and a footer with 'Web Server Definition and Plug-in' and '© 2005 IBM Corporation'.

The installation path is there so WebSphere can execute the Web Server programs, which are currently only important for the Apache-Based version IHS. The Configuration File Name presently defaults to <config root>/<AppServer name>, so you will probably want to change this. In this example, the path is into a <config root> tree for an application server. In the case of a Deployment Manager with a federated node, the add node process causes a glitch in a variable. You can edit the https.conf file by clicking the **Edit** or **Configuration File** file buttons. But there can be a problem when you save the file where both the changes and the original file are lost. Best practice is to keep a backup copy of the HTTP configuration file.

Process Definition

Web servers

Web servers > webserveronzDM > Log file > Plug-in properties > Process Definition

A process definition defines the command line information necessary to start/initialize a process.

Configuration

General Properties

Executable name
httpd

Executable arguments

startCommand
START WEBCAP

startCommandArgs

stopCommand
STOP WEBCAP

stopCommandArgs

Additional Properties

- Java Virtual Machine
- Environment Entries
- Process Execution

This field must not be empty in order for some of the other functions to work.

39

Web Server Definition and Plug-in © 2005 IBM Corporation

There is another problem in Web Server administration on z/OS. If the **Executable name** field is blank, it can cause error 500 when using other administrative functions. This field and all the fields on this screen are not used in the z/OS version, but there must be a value in this field.

Add the plug-in directives to the httpd.conf file

The best place to put the plug-in directives is beneath the comment in the httpd.conf file that says:

```
*** WAS Directives ***
```

Do not forget to comment out the `Pass /*` directive in the httpd.conf file; otherwise, the Service directive might not be processed (depending on the order of directives in the httpd.conf file).

Each of these directives must be entered entirely on one line. The lines are split on this slide due to space limitations.

```
ServerInit /usr/lpp/zWebSphere/V6R0/bin/ihs390WAS60Plugin_http.so:init_exit
          <very_long_pathname>plugin-cfg.xml

Service   /SuperSnoopWeb/*
          /usr/lpp/zWebSphere/V6R0/bin/ihs390WAS60Plugin_http.so:service_exit

ServerTerm /usr/lpp/zWebSphere/V6R0/bin/ihs390WAS60Plugin_http.so:term_exit
```

This slide demonstrates manually installing a plug-in configuration file. Adding the plug-in directives to the httpd.conf file is **not** done automatically. Be careful when performing this edit as it is easy to make a mistake. Include the full path to the plugin-cfg.xml file.

In this example, *SuperSnoopWeb* is the context-root of the application for which requests will be forwarded to WebSphere Application Server.

Here is the location of the plugin-cfg.xml file on the system:

```
/wasV6config/lwcell/lwnoded/AppServer/profiles/default/config/cells/lwcellnd/nodes/lwnode
d/servers/mywebserver/plugin-cfg.xml
```

The easiest way to find the plugin-cfg.xml file on your system is by issuing the find command from the Unix shell:

```
find . -name plugin-cfg.xml
```

The Web server must be stopped and restarted to process the ServerInit, Service, and ServerTerm directives. It must also be stopped and restarted if you make any changes to the plugin-cfg.xml file, since the ServerInit directive is executed only when the Web server is started. The Service directive is executed on each Web server request cycle, unless a preceding directive catches and processes the request first.

If you want all requests to be handled by the plug-in, then the first parameter of the Service directive can be specified as simply `/*`. If this technique is used, then be sure to specify that Service directive after all Pass, Exec, Fail, and Map directives in the httpd.conf file so that the Pass, Exec, Fail, and Map directives will also be processed.

Section

Summary and Reference

This section includes the summary and references.

Summary

- A new concept in unifying administration of Web Servers
- WebSphere Application Server V6 supports defining Web servers as a managed or unmanaged node
- Supports creating custom plug-in configuration files for each Web server



This presentation explained the enhancements in WebSphere Application Server V6 support for Web servers. The current implementation is aimed at Apache-based Web servers for distributed platforms. The presentation also detailed the concept of managed and unmanaged nodes for Web servers and explained the capabilities provided by the Apache-based IBM HTTP Server V6.

Reference

- **Configuring and Troubleshooting the WebSphere for z/OS Version 5 HTTP Server Plug-in**
 - ▶ <http://www.ibm.com/support/techdocs/atsmastr.nsf/WebIndex/PRS829>
- **IBM HTTP Server**
 - ▶ <HTTP://www-306.ibm.com/software/webservers/HTTPservers/library/>
- **Understanding the WebSphere Application Server Web server plug-in**
 - ▶ HTTP://www-106.ibm.com/developerworks/websphere/library/techarticles/0310_cocasse/cocasse.html



Shown here are some additional references.

Appendix

Wsadmin commands to create Web server definition

Appendix provides details of wsadmin tasks to create and manage Web server definitions.

Web Server Support in wsadmin Scripting

\$AdminTask:

- createWebServer –interactive
- Step 1 input
 - ▶ Node
 - ▶ Webserver name
 - ▶ Template
- Step 2 input
 - ▶ Port
 - ▶ webserverInstallRoot
 - ▶ **pluginInstallLocation**
 - ▶ configurationFile
 - ▶ errorLogfile
 - ▶ accessLogfile
 - ▶ serviceName (Windows only)
 - ▶ webserverProtocol (HTTP/HTTPS)
 - ▶ adminPort (remote IHS only)
 - ▶ adminProtocol
 - ▶ adminUserId
 - ▶ adminPassword
- deleteServer
- listWebServerTemplates - to display the new webserver templates

\$AdminConfig:

- ▶ remove
 - Input is configuration ID.
- ▶ Modify -
- ▶ types - webserver definition.
- ▶ show -
- ▶ showAttributes

\$AdminControl

- startServer – invoked thru Mbean using ProcessDefintion (server.xml)
 - ▶ StartCmd and StartCmdArgs.
- stopServer - – invoked thru Mbean using ProcessDefintion(server.xml)
 - ▶ StartCmd and StartCmdArgs.



wsadmin tasks to create web server and modify the configuration are shown here for reference.

configureWebserverDefinition Usage

```
wsadmin.sh -f \${WAS_HOME}/bin/configureWebserverDefinition.jacl
```

- `webserverName` : Name of the Web server
- `webserverType` : Type of the web server.
- `webserverInstallLocation` : Location of the Web server
- `webserverConfigFile` : Web server configuration file
- `webserverPort` : Listening port of the web server (80)
- `mapApplications` : Specifies whether the existing applications are to be mapped to the web server.
Valid values are: `MAP_NONE` `MAP_DEFAULT` `MAP_ALL`
- `pluginInstallLocation` : Location of the plug-in install
- `webserverNodeType` : Type of the node holding Web server
Valid values are : `managed`, `unmanaged`
- `webserverNodeName` : Node at which web server should be defined (created unmanaged if does not exist)
- `webserverHostName` : Host name of the web server (Needed only for unmanaged node)
- `webserverOS` : Operation system of the web server machine (Needed only for unmanaged node)
Values are : `windows` `solaris` `aix` `hpux` `linux` `os400` `os390`



wsadmin tasks to create web server definition, using `configureWebserverDefinition.jacl` script is shown here for reference.

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
eIogo business	DB2	Series	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 2004. 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.

