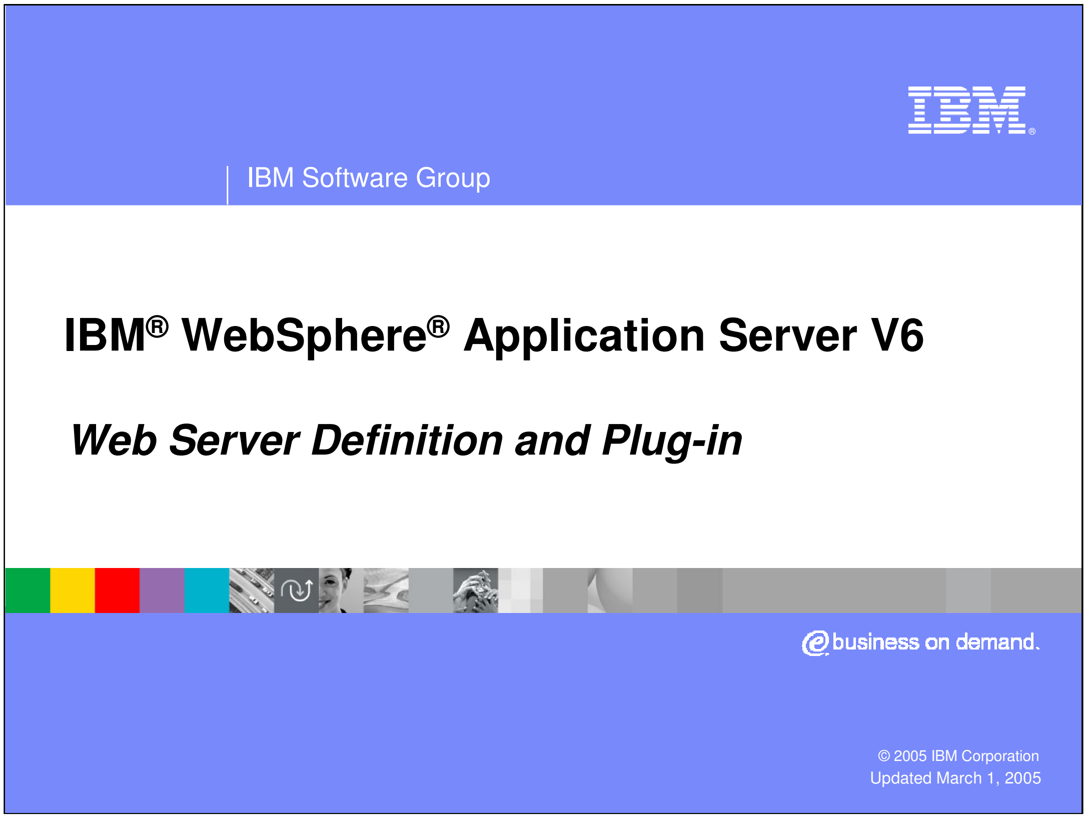


IBM Software Group

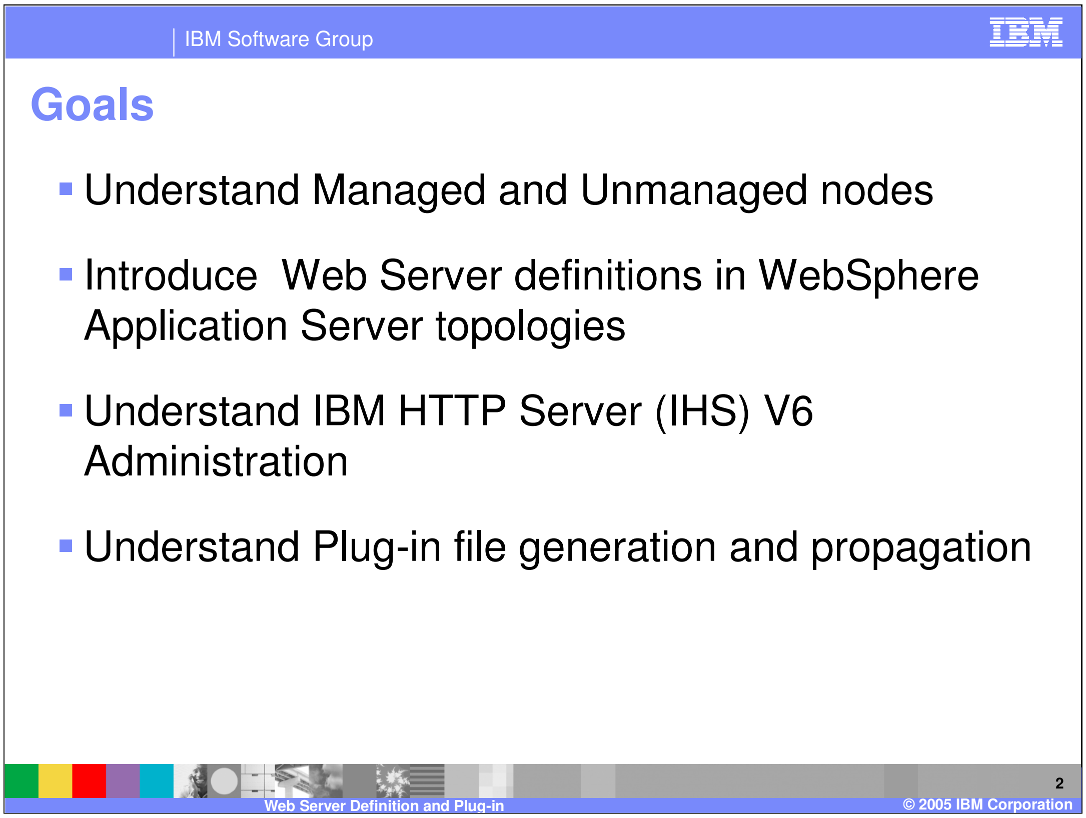
IBM® WebSphere® Application Server V6

Web Server Definition and Plug-in

business on demand.

Updated March 1, 2005

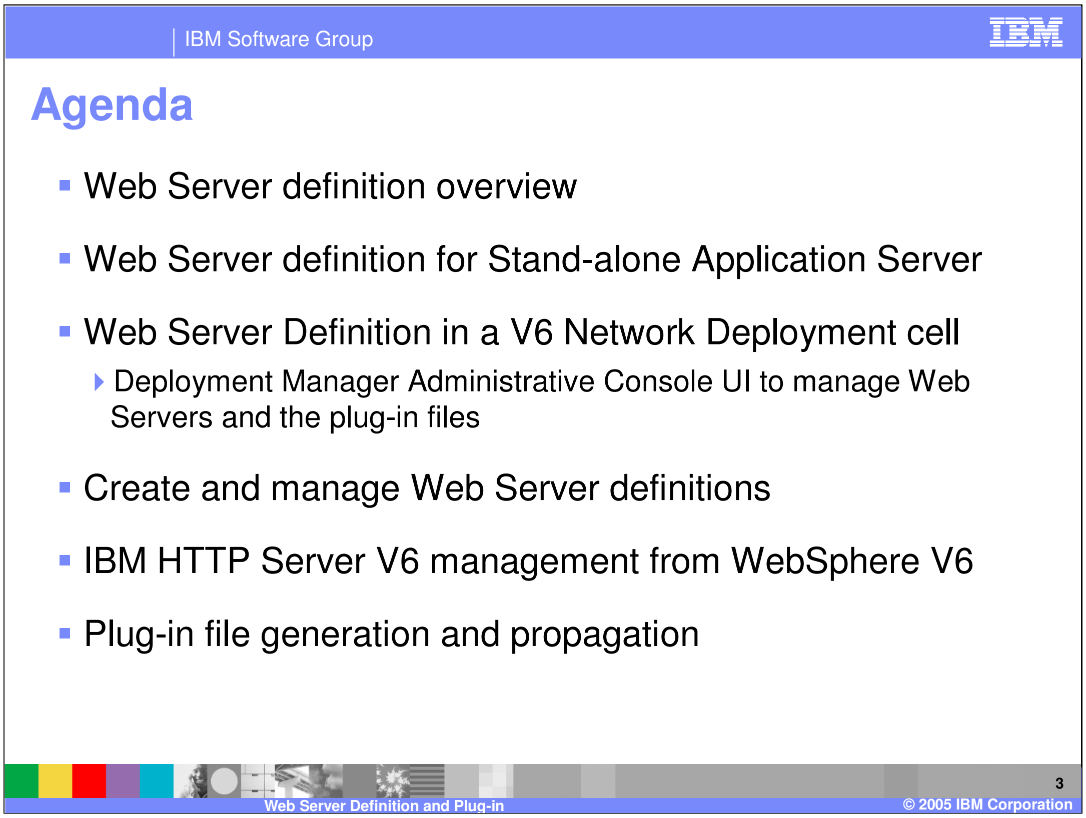
This presentation will focus on the support for Web Servers in WebSphere Application Server V6.



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

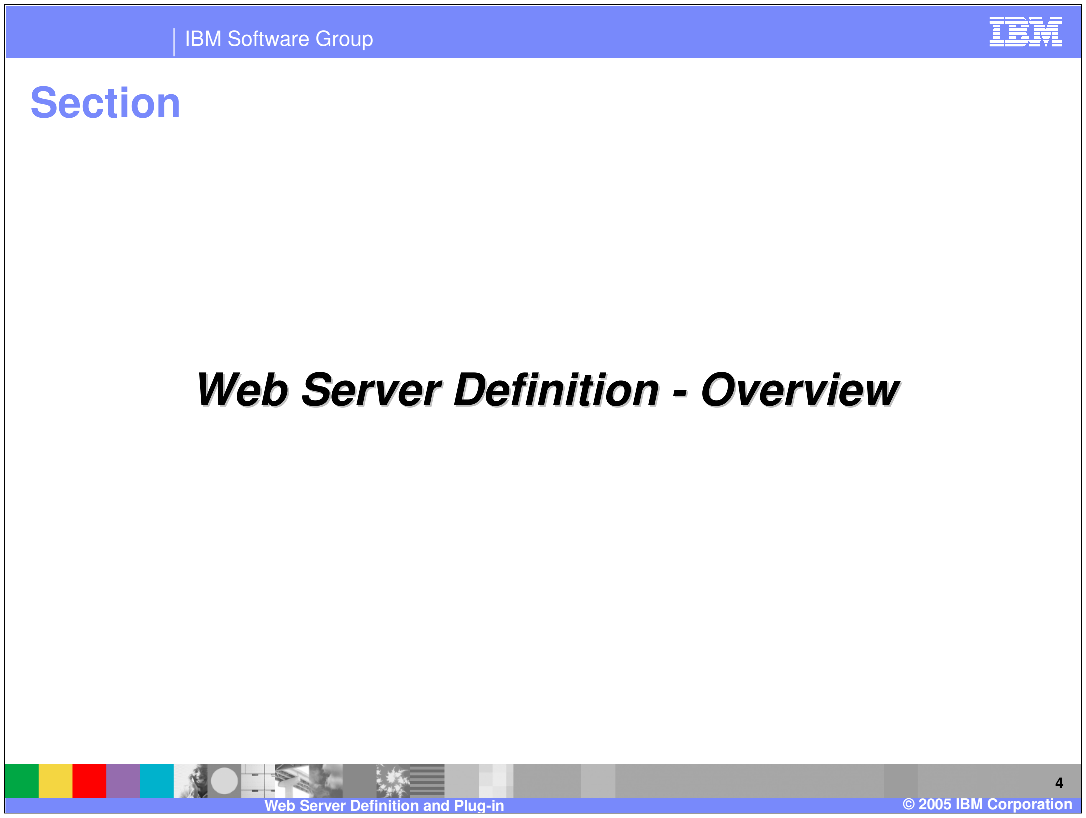
The goal of this presentation is to introduce the core concepts of Web Servers in WebSphere Application Server V6. In particular this presentation will cover the new concept of managed and unmanaged nodes within a WebSphere topology. There will also be an explanation of Web Server administrative tasks.



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

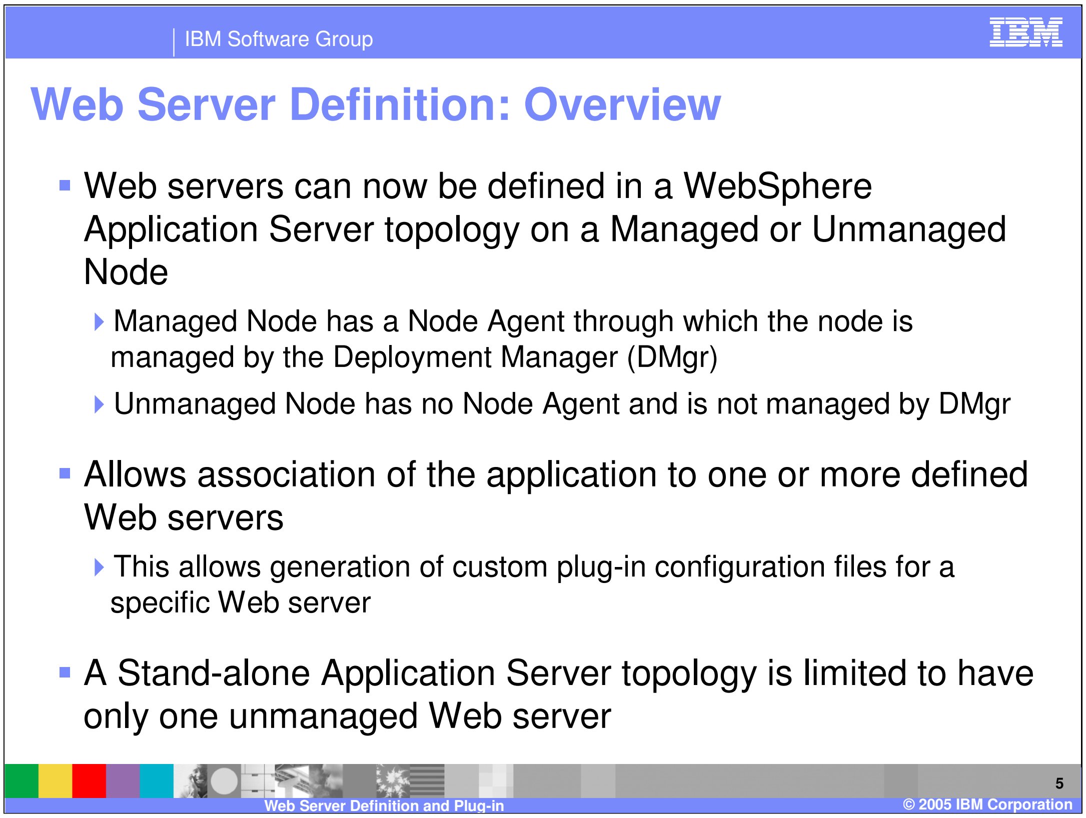
This presentation will begin by explaining how to create a Web Server definition in an ND cell. In V6 there is an option to either create a managed or unmanaged node, for the purposes of managing the Web Server from within a WebSphere topology. The WebSphere administration console provides the capabilities to manage the Web Servers and their plug-in files. After that detail, the presentation will show how to create a Web Server in a Base or Express environment. This will cover in details managing IBM HTTP Server through WebSphere. The presentation will conclude with Plug-in file generation and propagation features in WebSphere Application Server V6.



Section

Web Server Definition - Overview

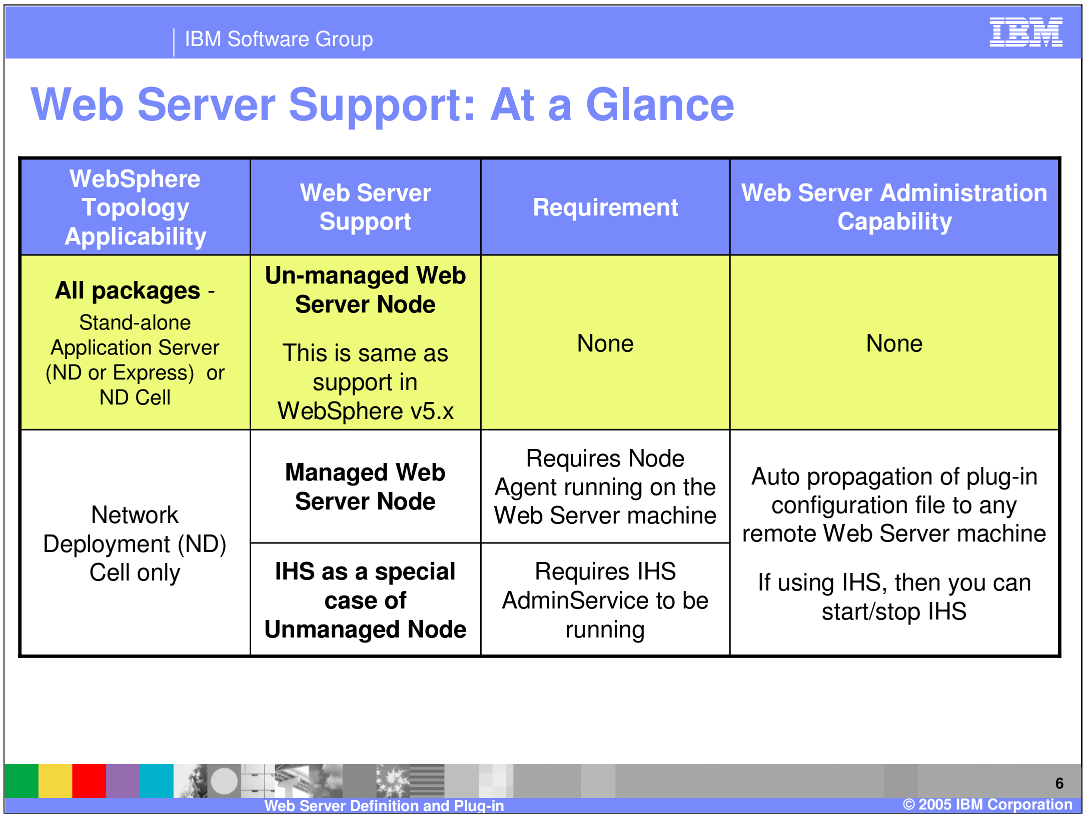
The next section will cover the basics of Web Server definitions.



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
- Allows association of the application to 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 have only one unmanaged Web server

New in V6 is the ability to define a web server configuration within a node. In the case of the IBM HTTP Server, the WebSphere administrative console can stop and start the web server, and transparently update the plug-in configuration file on a remote machine where the web server is installed. Also, WebSphere can selectively target applications to specific web servers, so that only specific web servers can route requests to the 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 same system as the Web Server. With an unmanaged node there is no node agent, this can be useful if you want to place the Web Server in a DMZ.



IBM Software Group

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 IHS AdminService to be running	If using IHS, then you can start/stop IHS

Web Server Definition and Plug-in

6

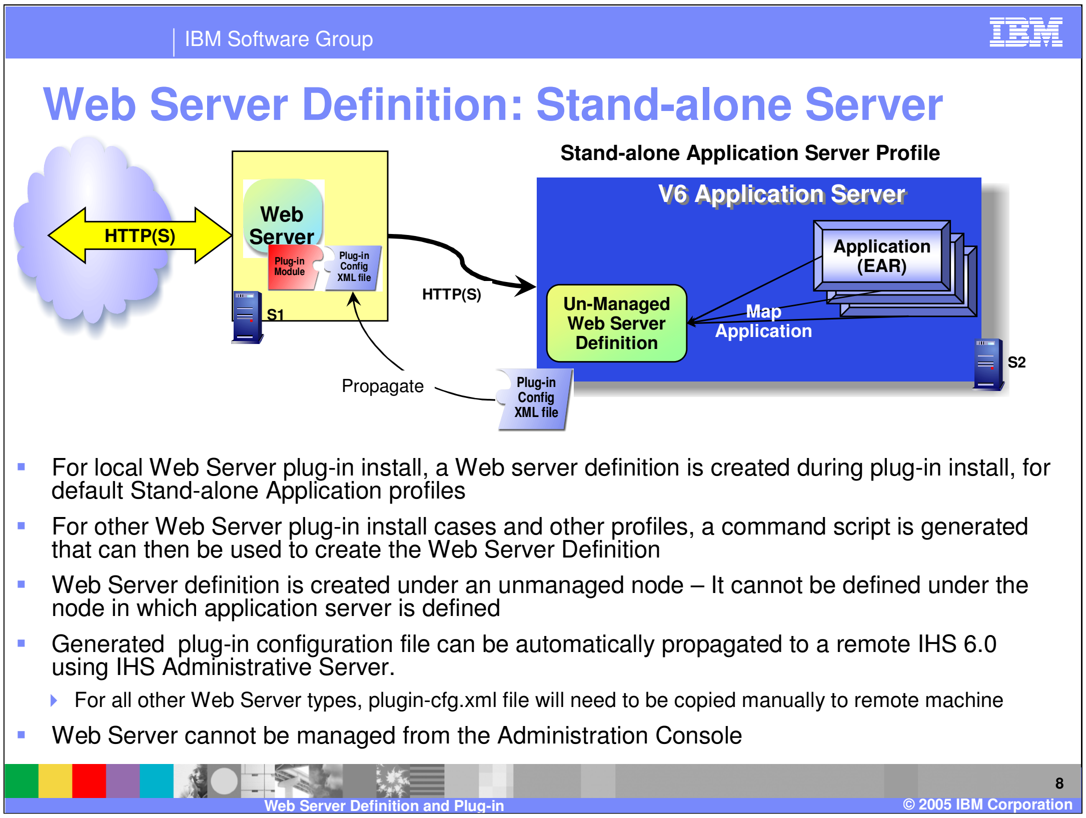
© 2005 IBM Corporation

This slide details the support for managed and unmanaged nodes based on the version of WebSphere you use. 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

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



IBM Software Group

Web Server Definition: Stand-alone Server

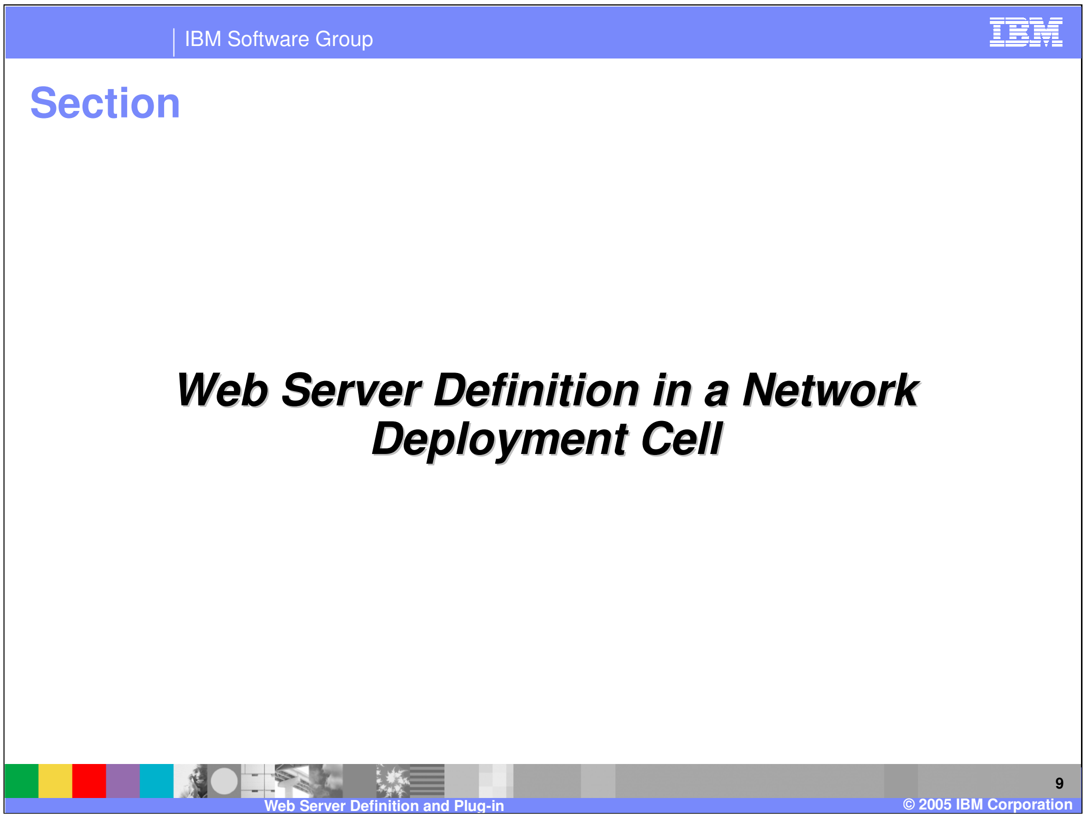
- For local Web Server plug-in install, a Web server definition is created during plug-in install, for default Stand-alone Application profiles
- For other Web Server plug-in install cases and other profiles, a command script is generated that can then be used to create the Web Server Definition
- Web Server definition is created under an unmanaged node – It cannot be defined under the node in which application server is defined
- Generated plug-in configuration file can be automatically propagated to a remote IHS 6.0 using IHS Administrative Server.
  - 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 Administration Console

8

Web Server Definition and Plug-in

© 2005 IBM Corporation

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 will have to be propagated to the server that the Web Server is installed on. Propagation is supported for the IHS 6.0 web server using IHS administration server. WebSphere V6 as the ability to map applications to specific Web Servers, in this case, since there can be only a single web server, all applications will be mapped to the same Web Server.



IBM Software Group

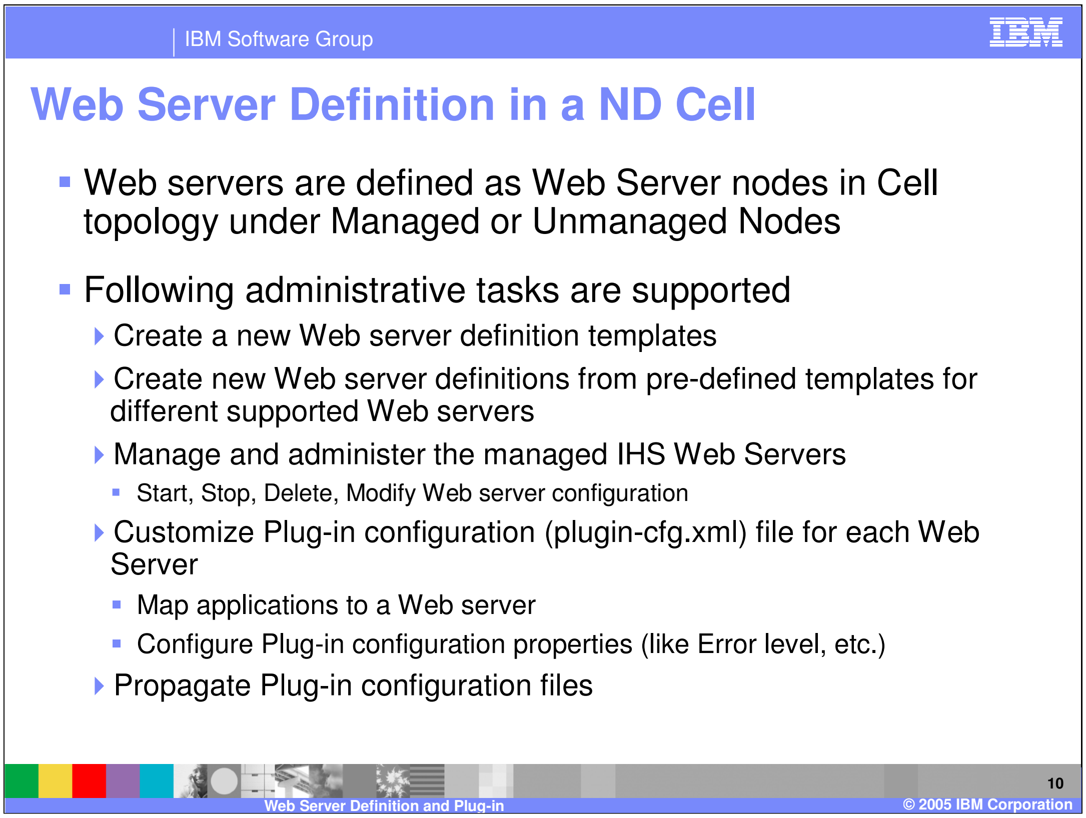
Section

Web Server Definition in a Network Deployment Cell

9

Web Server Definition and Plug-in

The next section will detail a web server definition in a Network Deployment setting.



IBM Software Group

Web Server Definition in a ND Cell

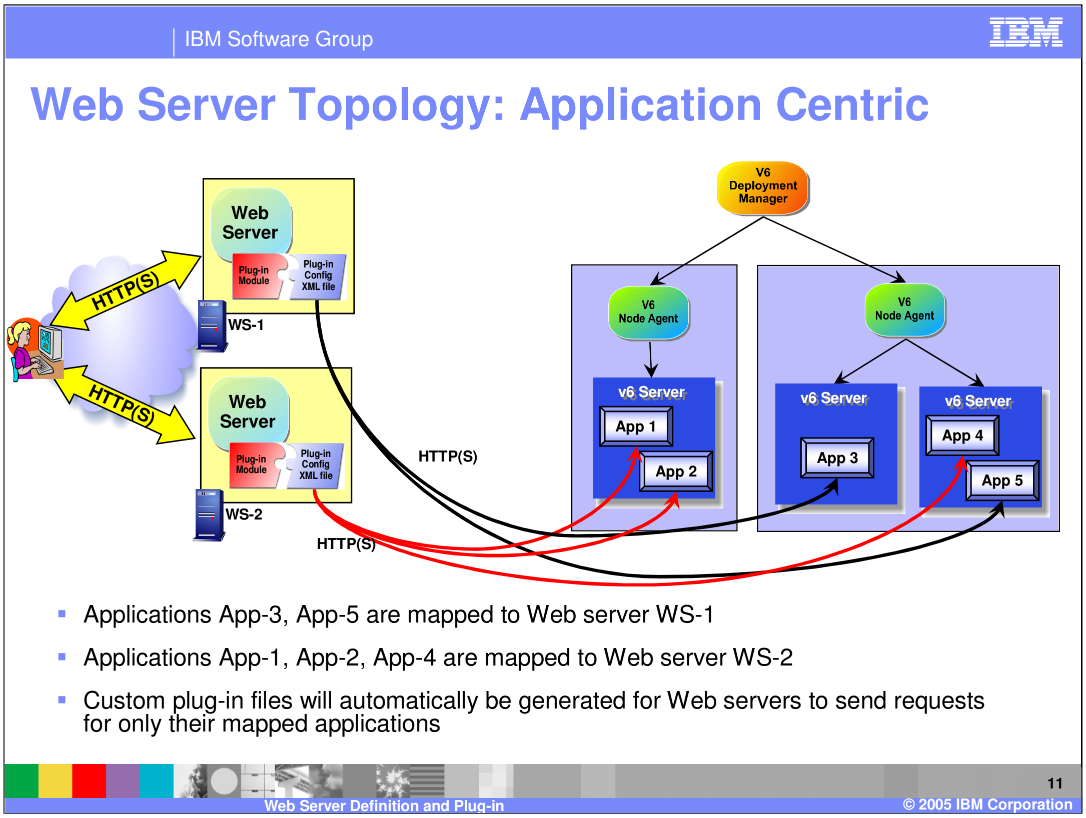
- Web servers are defined as Web Server nodes in Cell topology under Managed or Unmanaged Nodes
- Following administrative tasks are supported
  - Create a new Web server definition templates
  - Create new Web server definitions from pre-defined templates for different supported Web servers
  - Manage and administer the managed IHS 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

Web Server Definition and Plug-in

10

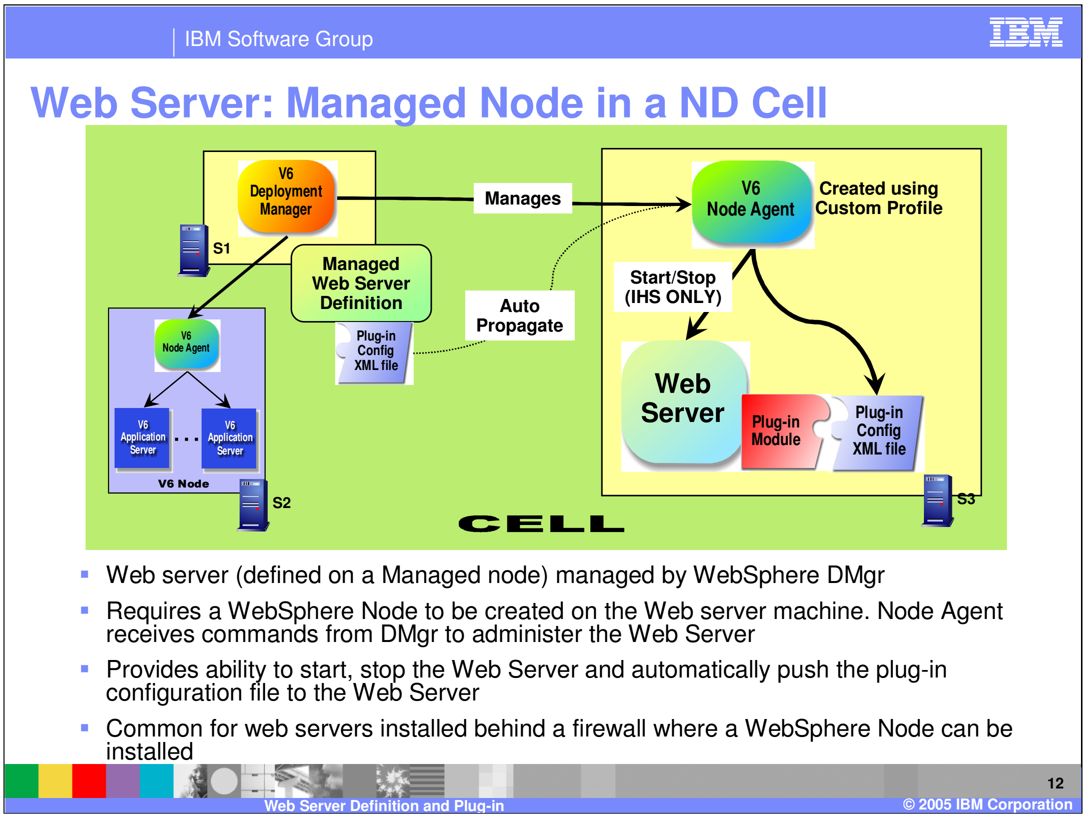
In a Network Deployment environment, Web Servers can either be define as a managed or unmanaged node. As a managed node, there will be a node agent present that will allow the deployment manager to manage the web server. In an unmanaged node there is no node agent, so the deployment manager is more limited in dealing with the web server. Administrative tasks allow a user 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 propagated to managed web servers, or manually copied to unmanaged 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

V6 Deployment Manager

S1

Manages

V6 Node Agent

Created using Custom Profile

Managed Web Server Definition

Plug-in Config XML file

Auto Propagate

Start/Stop (IHS ONLY)

V6 Node Agent

V6 Application Server

. . .

V6 Application Server

V6 Node

S2

Web Server

Plug-in Module

Plug-in Config XML file

S3

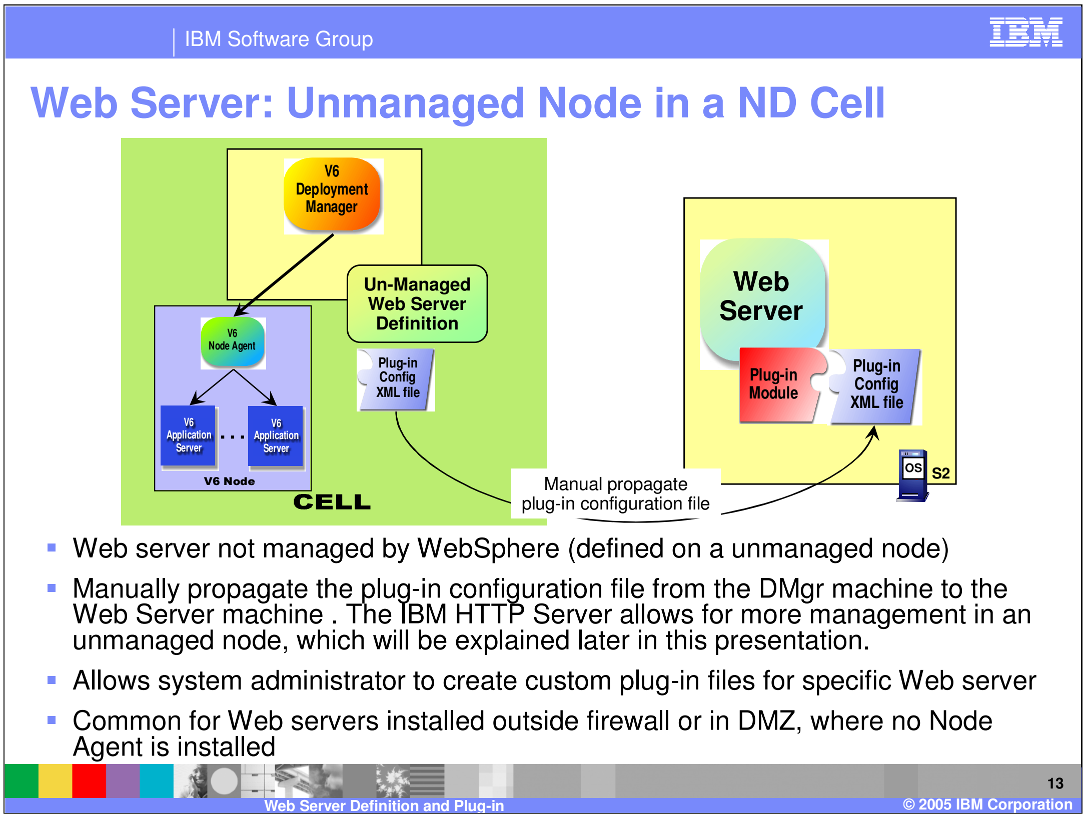
CELL

- Web server (defined on a Managed node) managed by WebSphere DMgr
- Requires a WebSphere Node to be created on the Web server machine. Node Agent receives commands from DMgr to administer the Web Server
- Provides ability to start, 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 server where the web server is installed. The web server is managed by the WebSphere Deployment Manager through the node agent. The node agent provides the ability to start and stop the Web Server as well as automatically push the plug-in configuration file to the Web Server. This scenario is most common for behind a firewall where a WebSphere Node can be installed without any security concerns.

IBM Software Group

Web Server: Unmanaged Node in a ND Cell

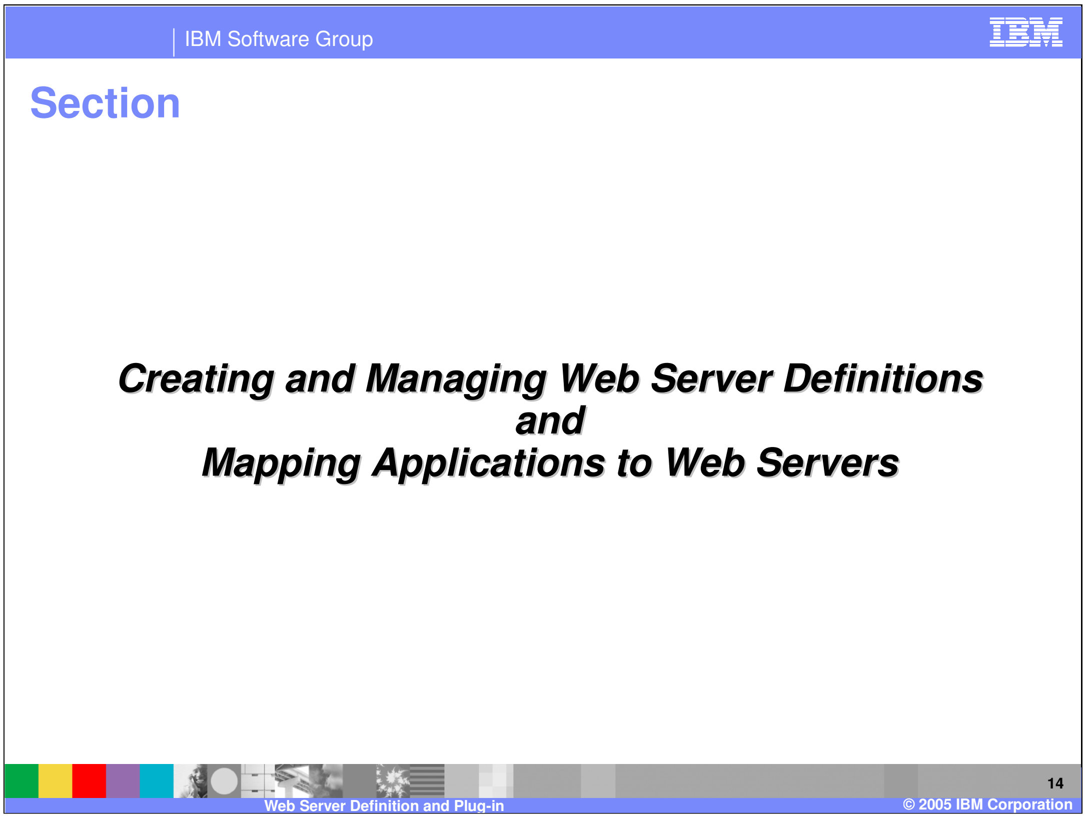


- Web server not managed by WebSphere (defined on a unmanaged node)
- Manually propagate the plug-in configuration file from the DMgr machine to the Web Server machine . The IBM HTTP Server allows for more management in an unmanaged node, which will be explained later in this presentation.
- Allows system 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

Web Server Definition and Plug-in

13

In this example the web server is defined in an unmanaged node. The Web Server is registered as 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. The IBM HTTP Server allows for more management in an unmanaged node, this will be explained later in this presentation.



IBM Software Group

Section

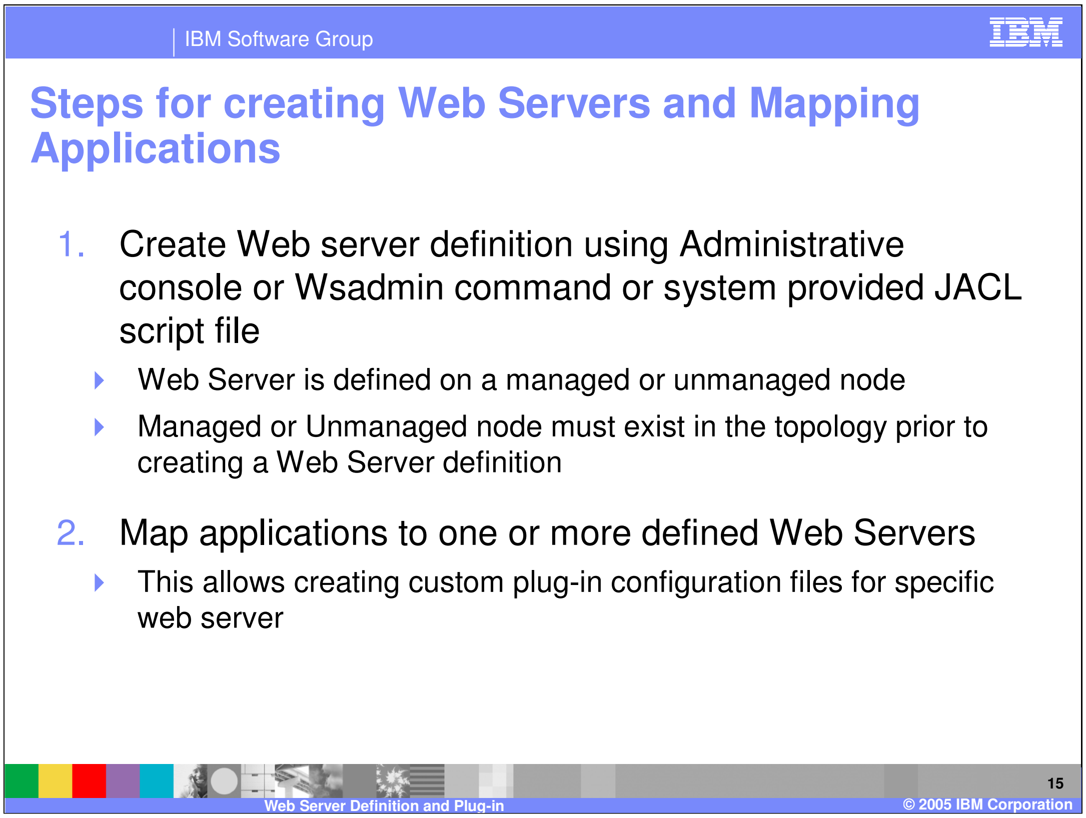
Creating and Managing Web Server Definitions
and
Mapping Applications to Web Servers

14

Web Server Definition and Plug-in

© 2005 IBM Corporation

The next section will show how create and manage web server definitions .



IBM Software Group

Steps for creating Web Servers and Mapping Applications

1. Create Web server definition using Administrative console or Wsadmin command or system provided JACL script file
   - Web Server is defined on a managed or unmanaged node
   - Managed or Unmanaged node must exist in the topology prior to creating a Web Server definition
2. Map applications to one or more defined Web Servers
   - This allows creating custom plug-in configuration files for specific web server

Web Server Definition and Plug-in

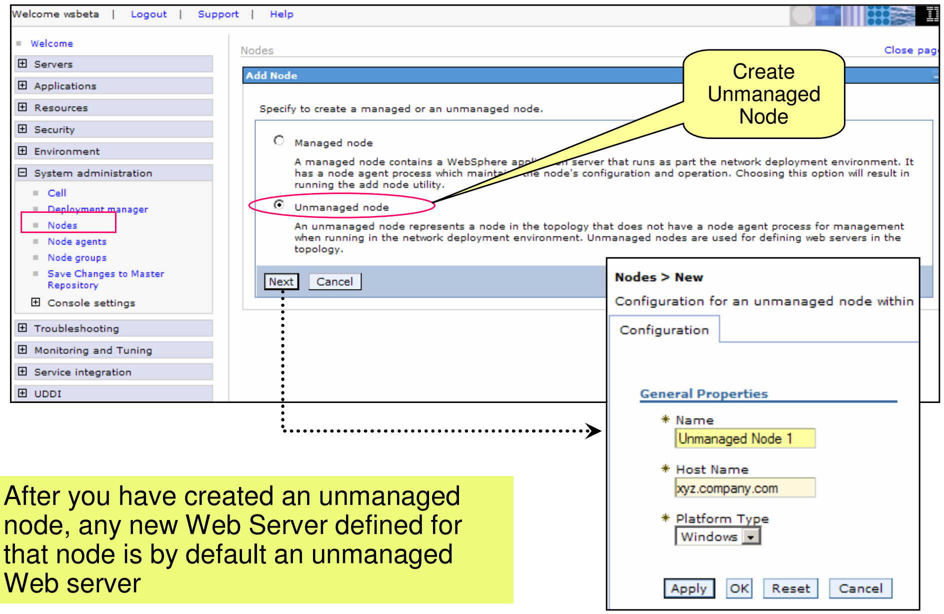
15

Subsequent foils will go into the details of each of these steps.

Web Server is defined on either a managed or unmanaged node. One can use the Administrative console, or wsadmin commands to create the Web server definition. In addition, there is a JACL script called configureWebserverDefinition, that can be used by wsadmin to create the Web server definition.

Once the Web servers are defined, the user applications are mapped to the web servers. This allows creating the custom plug-in files for specific web servers, containing only those applications that are mapped to the web server. It is possible to have an application map to multiple web servers. In that case, the plug-in files for those web servers will contain entries for the application. This allows multiple web servers to be used for the same application.

Creating Unmanaged Node for Web Server



Specify to create a managed or an unmanaged node.

○ Managed node

A managed node contains a WebSphere application server that runs as part the network deployment environment. It has a node agent process which maintains the node's configuration and operation. Choosing this option will result in running the add node utility.

● Unmanaged node

An unmanaged node represents a node in the topology that does not have a node agent process for management when running in the network deployment environment. Unmanaged nodes are used for defining web servers in the topology.

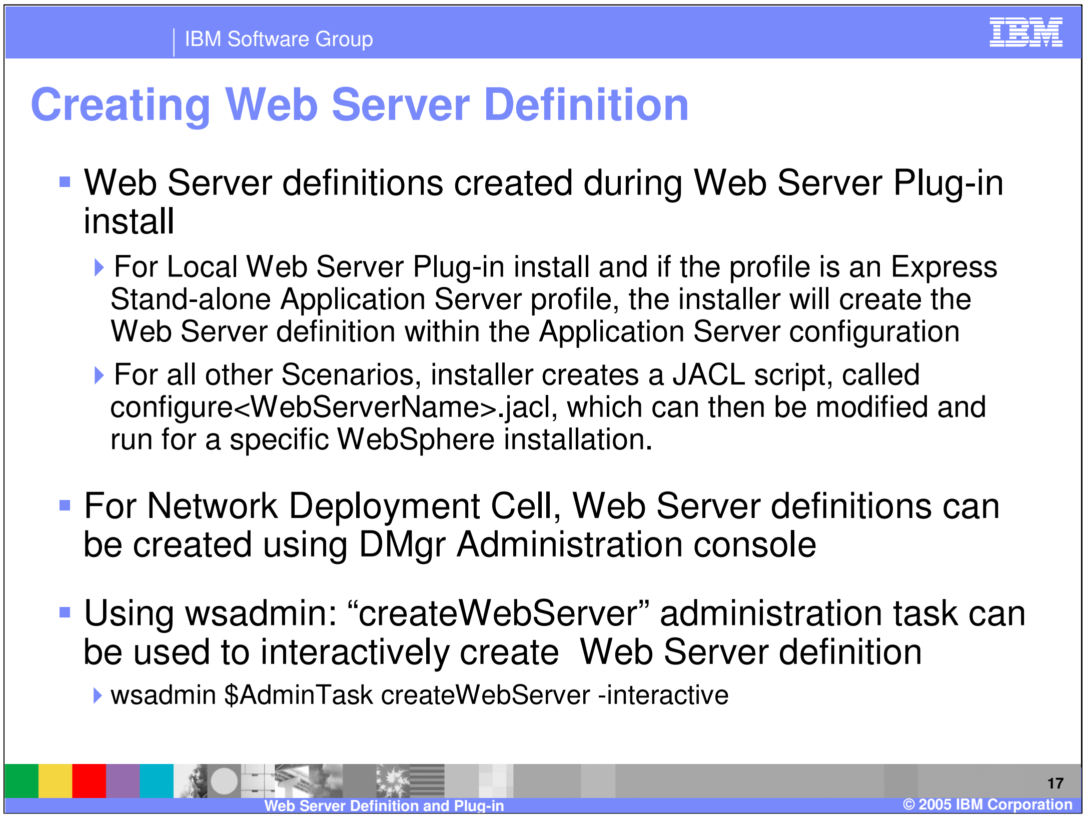
Configuration for an unmanaged node within

* Name	Unmanaged Node 1
* Host Name	xyz.company.com
* Platform Type	Windows

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

From the system administration panel, a new node can be created. This node can be either managed or unmanaged. The panels here shows how to create a unmanaged node.

Managed Node is created either through this panel, or when creating a Custom profile, or when federating a Stand-alone application server.

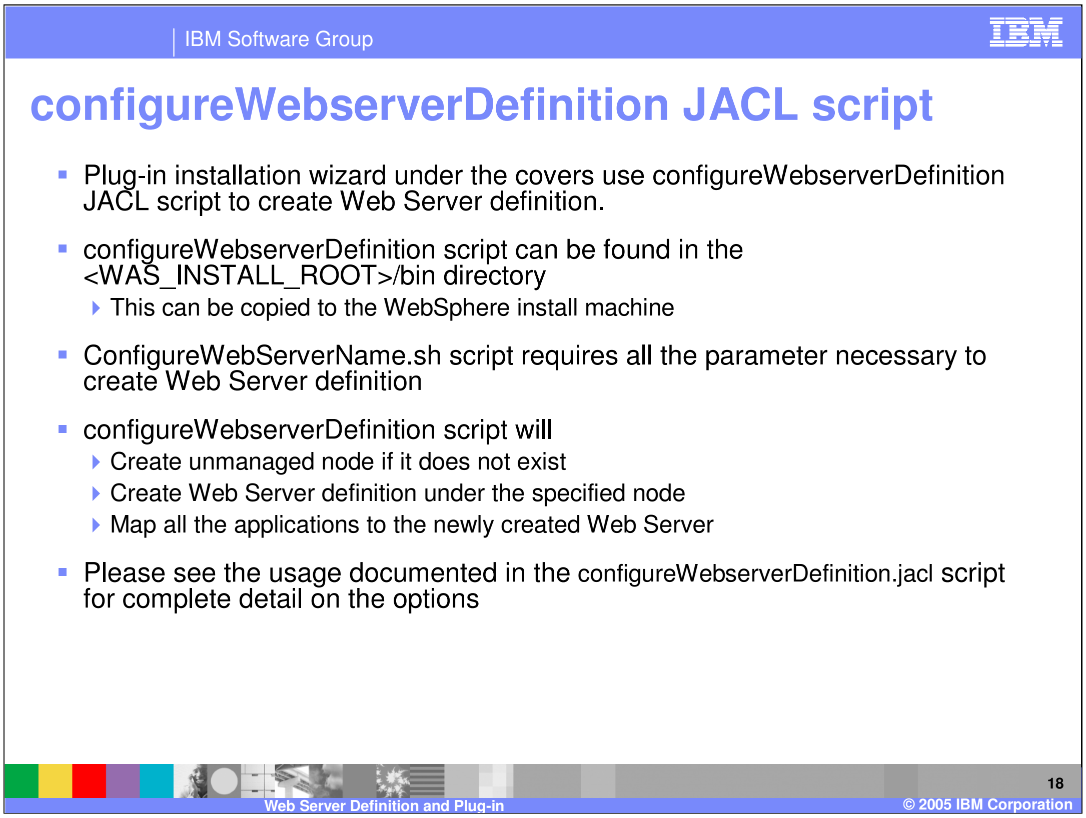


Creating Web Server Definition

- Web Server definitions created during Web Server Plug-in install
  - 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 Administration console
- Using wsadmin: “createWebServer” administration task can be used to interactively create Web Server definition
  - wsadmin $AdminTask createWebServer -interactive

Web Server definitions can be created during Plug-in install, or Administration, Console in Network Deployment environment or using command line wsadmin tool.

The plug-in install either creates the Web Server definition or create a JACL script, called configureWebserverDefinition.jacl, that enables the users to later create the Web Server definition. The JACL script is useful for remote Web Server plug-in, where the installer may not be able to access the WebSphere installed on some remote machine.



IBM Software Group

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

Web Server Definition and Plug-in

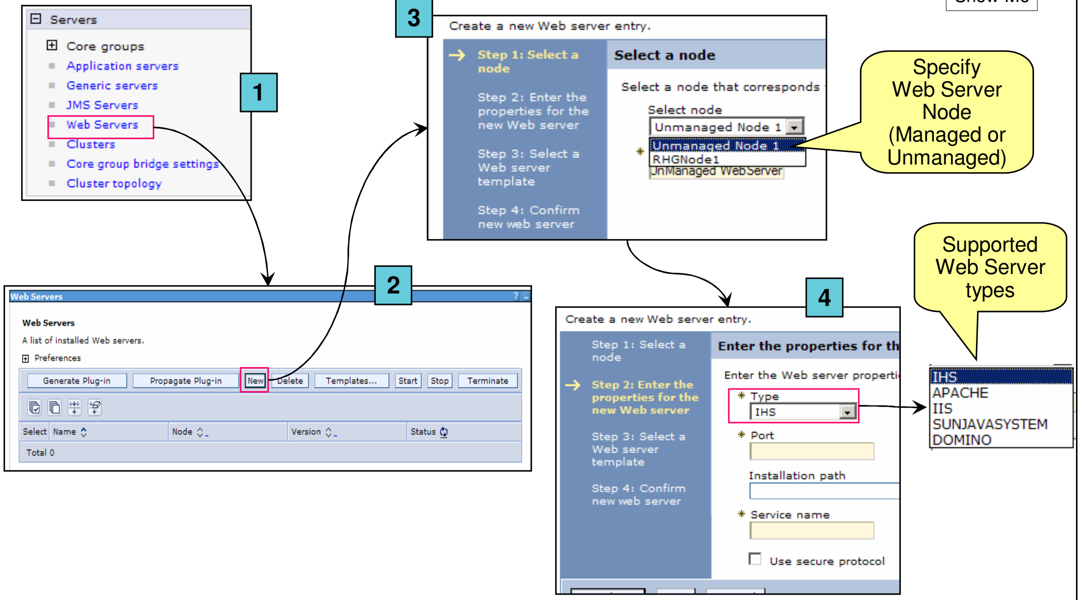
18

© 2005 IBM Corporation

Plug-in installer wizard under the covers uses configureWebserverDefinition JACL script to create Web Server definition. This foil describes the details of the JACL script. The script is saved in the Web Server plug-in install bin directory

Create/Manage New Web Server Definition

Show-Me



Web Servers

A list of installed Web servers.

⊞ Preferences

Select	Name	Node	Version	Status
Total 0				

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 1, RHGNode1, UnManaged WebServer)

Specify Web Server Node (Managed or Unmanaged)

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 properties for th

Enter the Web server properti

* Type: IHS

* Port

Installation path

* Service name

□ Use secure protocol

Supported Web Server types

The steps of creating a web server definition using 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.

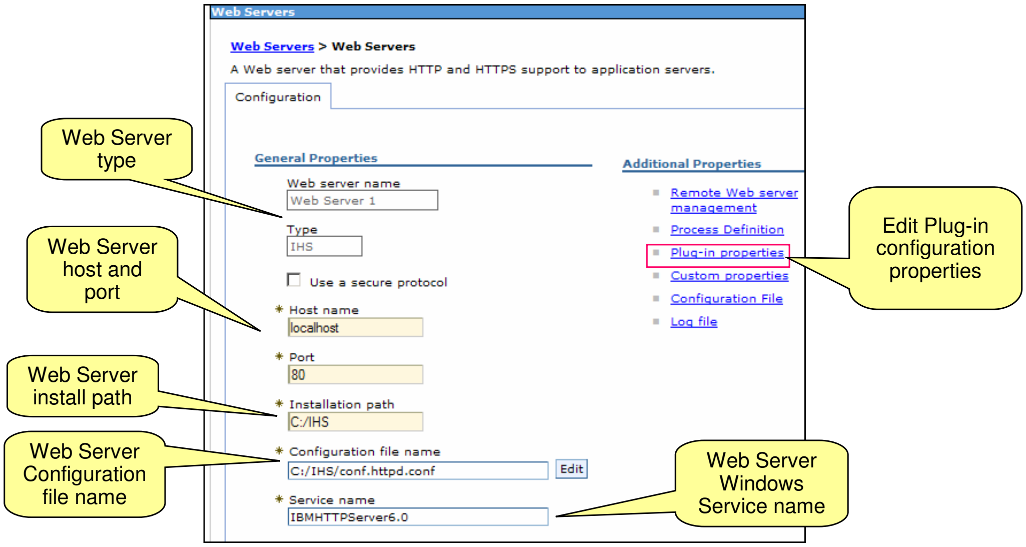
In Panel (3), the managed or unmanaged node is selected on which the web server will be defined. In the example show here, there are two possible node, both a managed and an unmanaged node have already been created in this topology.

After a node is selected, the properties for the new web server are entered. In panel (4), details of the web server is 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 [Edit]

* Service name: IBMHTTPServer6.0

Additional Properties

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

Web Server type

Web Server host and port

Web Server install path

Web Server Configuration file name

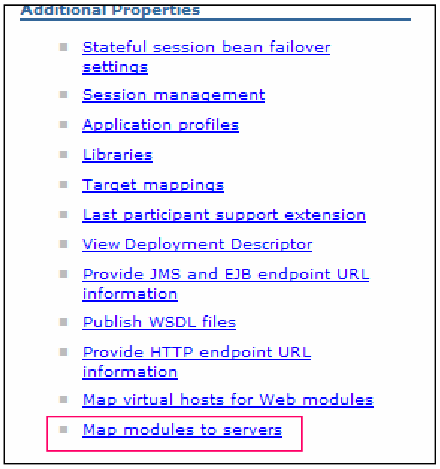
Web Server Windows Service name

Edit Plug-in configuration properties

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
- Next select the Application Server and one or more Web servers
- Also done via 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.

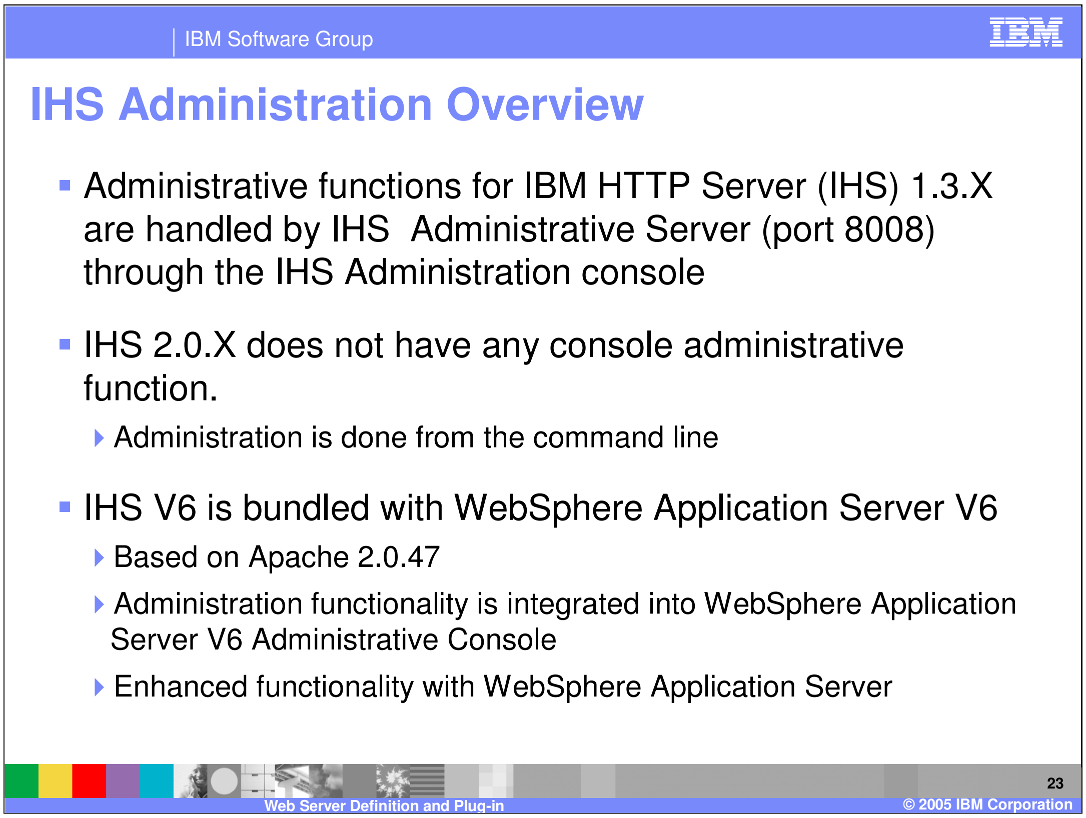
Section

WebSphere Application Server V6

and

IBM HTTP Server (IHS) V6

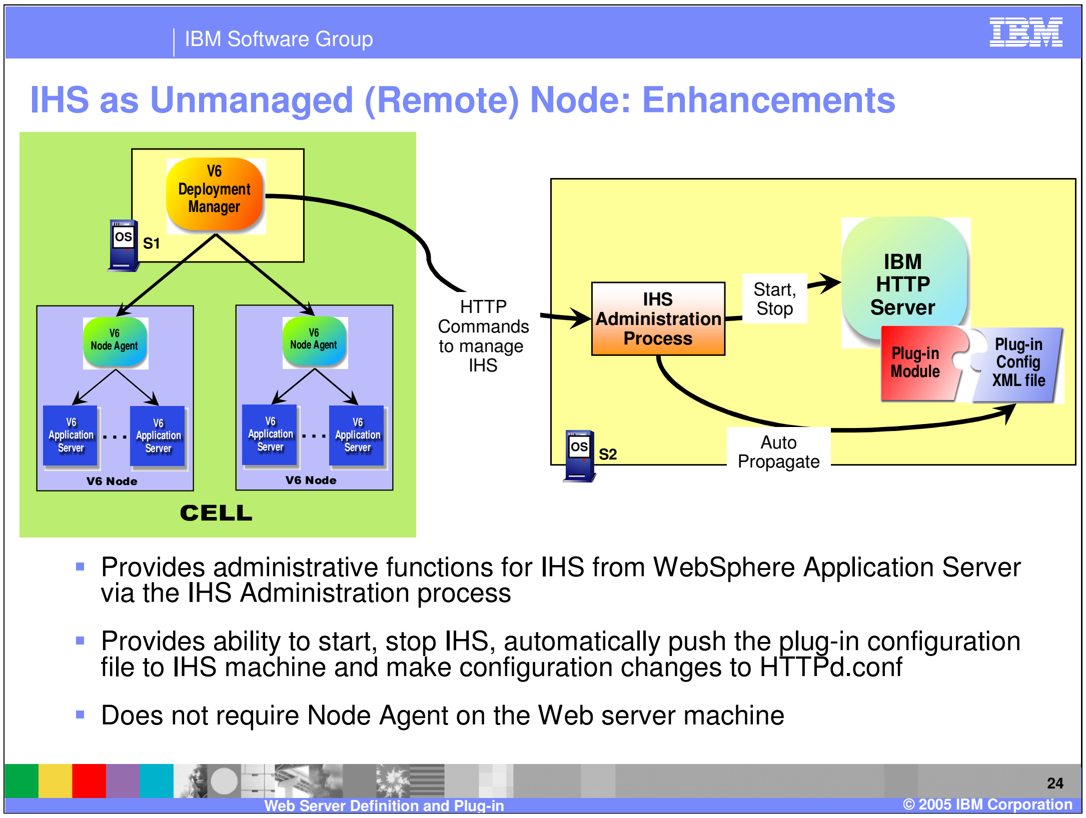
The next section will detail the enhancements for the IBM HTTP Server in V6.



IHS Administration Overview

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

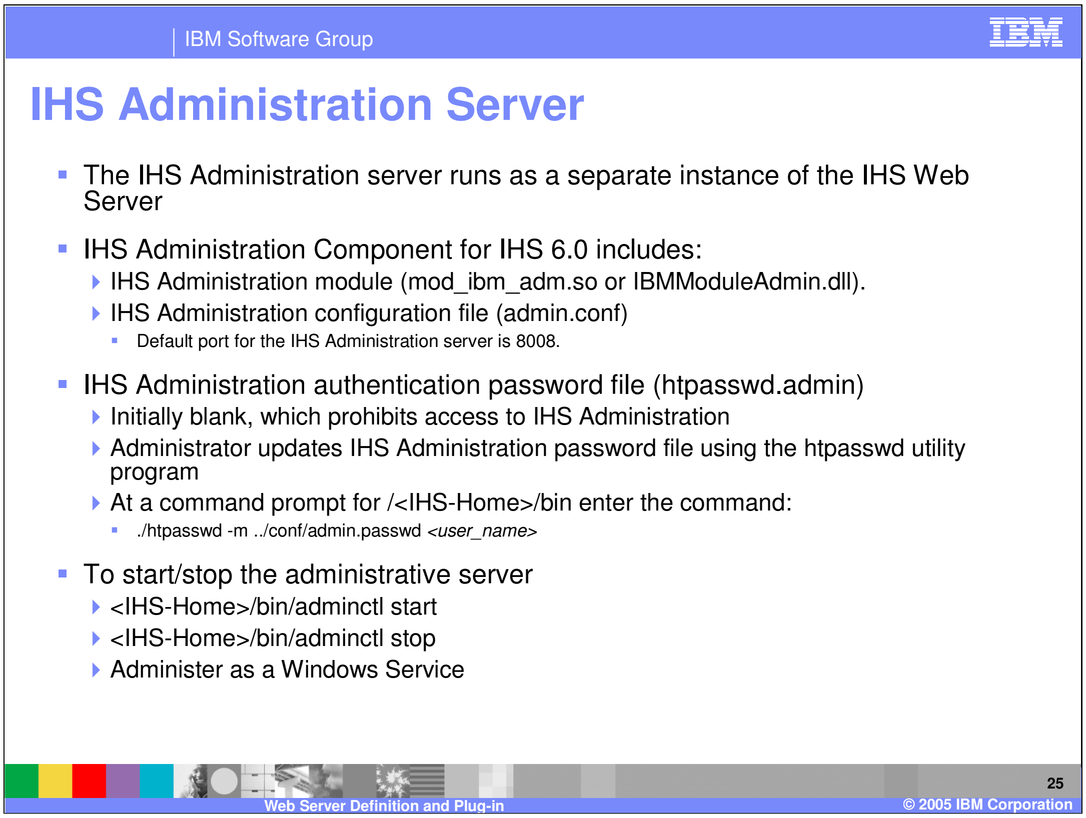
Earlier version of the IHS were administered through a separate administrative console. Since version 2.0.x of the IBM HTTP Server this has no longer been the case. Instead administration has been done from the command line. IHS V6 is bundled with WebSphere V6, more specific information on IHS can be found in the separate presentation for IHS.



IHS as Unmanaged (Remote) Node: Enhancements

- Provides administrative functions for IHS from WebSphere Application Server via the IHS Administration process
- Provides ability to start, stop IHS, automatically push the plug-in configuration file to IHS machine and make configuration changes to HTTPd.conf
- Does not require Node Agent on the Web server machine

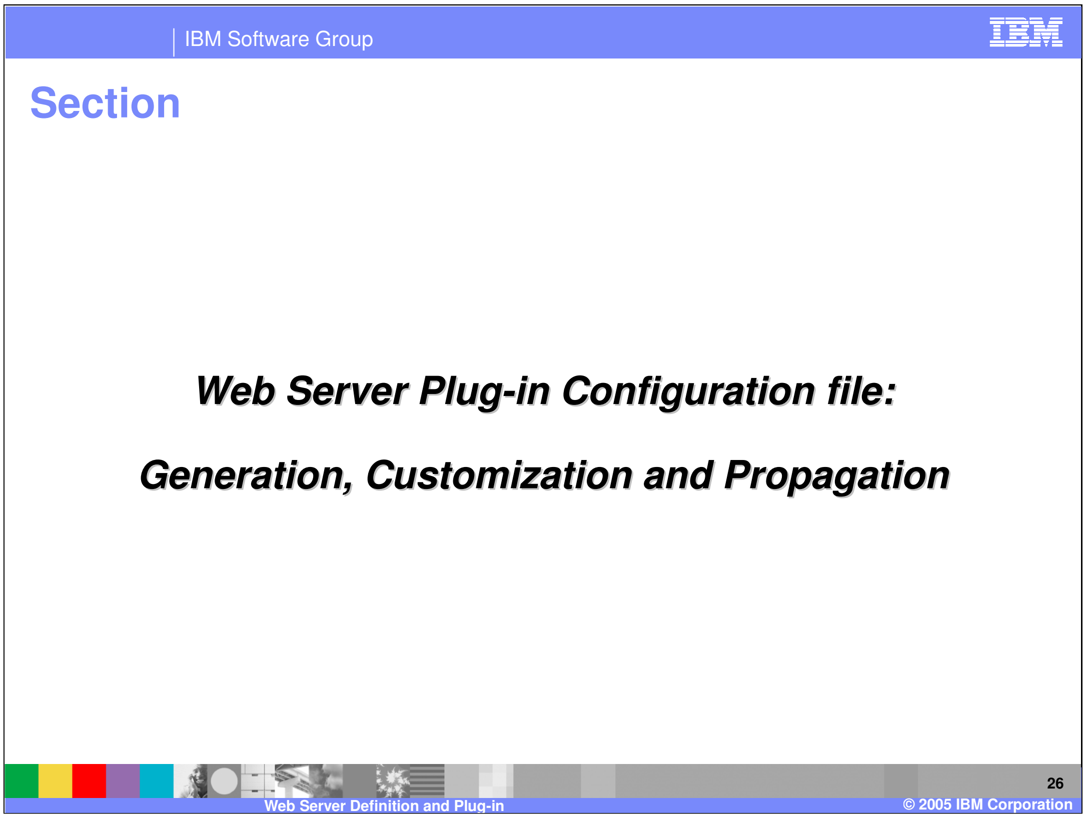
IBM HTTP Server provides enhanced functionality to manage an IHS running in an unmanaged node. This is accomplished through the use of a separate IHS Administration process that runs on the same server as the IHS server. The administration process runs as a separate process on the same server. The Deployment Manager can communicate with the administration process to manage the IHS server.



IHS Administration Server

- The IHS Administration server runs as a separate instance of the IHS Web Server
- IHS Administration Component for IHS 6.0 includes:
  - IHS Administration module (mod_ibm_adm.so or IBMModuleAdmin.dll).
  - IHS Administration configuration file (admin.conf)
    - Default port for the IHS Administration server is 8008.
- IHS Administration 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/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 administer the IBM HTTP Server. This is offered here as a reference.



IBM Software Group

Section

Web Server Plug-in Configuration file:

Generation, Customization and Propagation

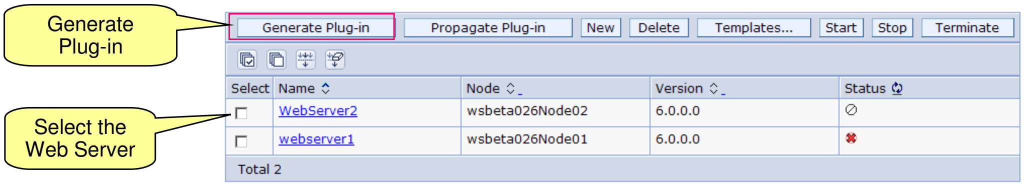
Web Server Definition and Plug-in

26

© 2005 IBM Corporation

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

Generating Plug-in with Administrative Console

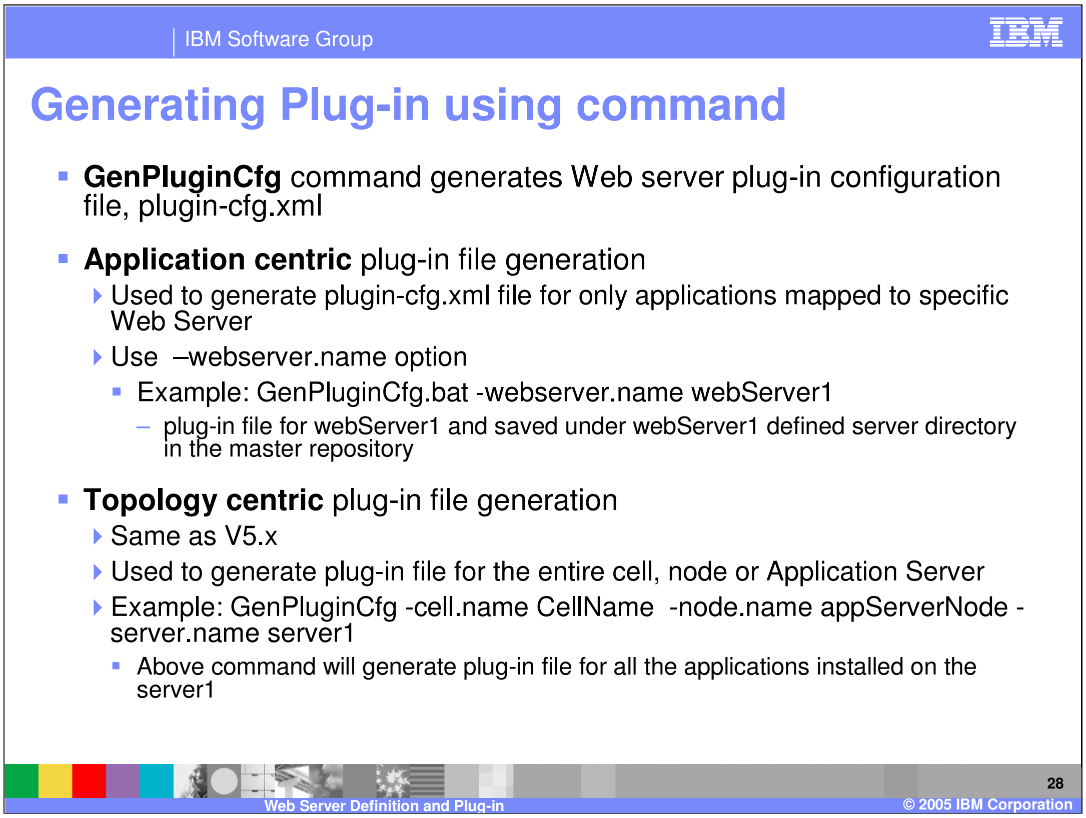


Select	Name	Node	Version	Status
□	WebServer2	wsbeta026Node02	6.0.0.0	⊘
□	webserver1	wsbeta026Node01	6.0.0.0	✖

Total 2

- Plug-in configuration file can be generated using Administration Console after defining Web Server and mapping the Applications.
- Only Applications mapped to the Web Server will be added 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 plug-in for that web server 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** command generates Web server plug-in configuration file, plugin-cfg.xml
- **Application centric** plug-in file generation
  - Used to generate plugin-cfg.xml file for only 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

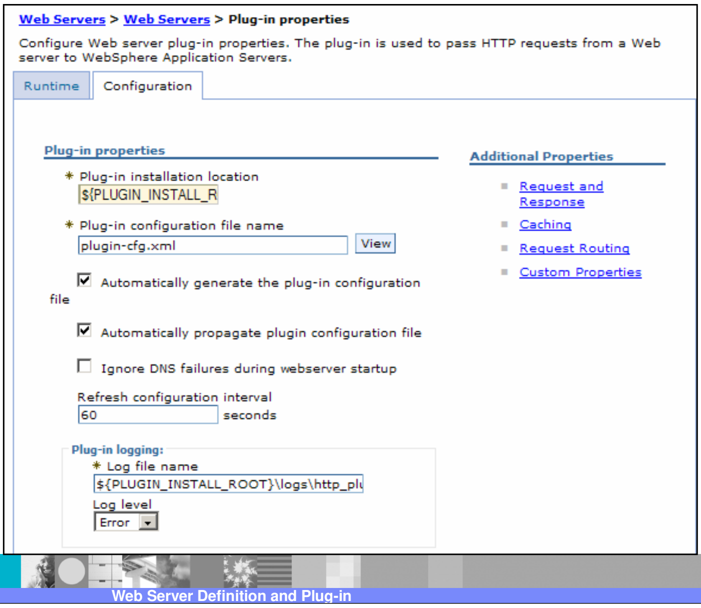
GenPluginCfg command is used to regenerate the WebSphere Web server plug-in configuration file, plugin-cfg.xml.

When the GenPluginCfg command is issued with the option -webserver.name webservrName, plug-in configuration file is created for the Web server. This settings 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 option -webserver.name webservrName, the plug-in configuration file is generated based on topology.

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.

Plug-in properties

* Plug-in installation location
${PLUGIN_INSTALL_R

* Plug-in configuration file name
plugin-cfg.xml

☑ Automatically generate the plug-in configuration file

☑ Automatically propagate plugin configuration file

☐ Ignore DNS failures during webserver startup

Refresh configuration interval
60 seconds

Plug-in logging:

* Log file name
${PLUGIN_INSTALL_ROOT}\logs\http_plu

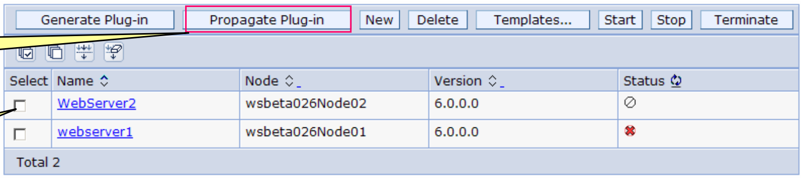
Log level
Error

Additional Properties

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

This is the screen to configure the plug-in in the administrative console. 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



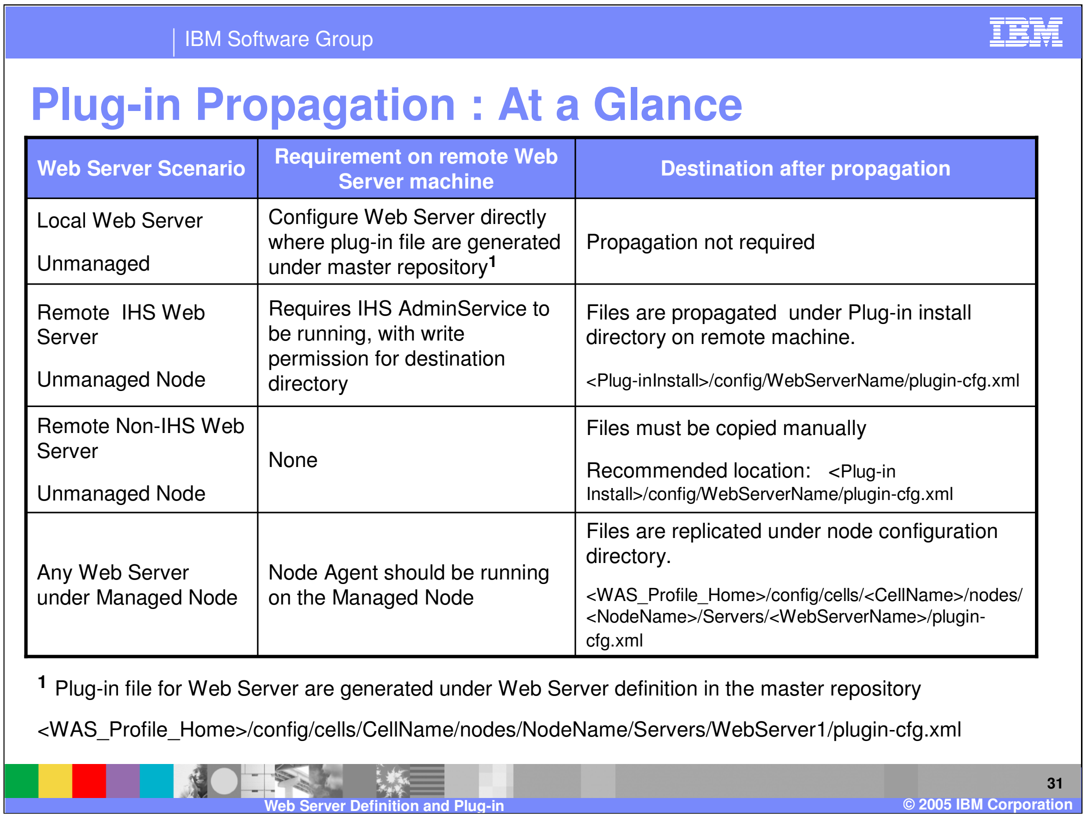
Select	Name	Node	Version	Status
□	WebServer2	wsbeta026Node02	6.0.0.0	⊘
□	webserver1	wsbeta026Node01	6.0.0.0	✖

Total 2

Propagate Plug-in

Select the Web Server

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



IBM Software Group

Plug-in Propagation : At a Glance

Web Server Scenario	Requirement on remote Web Server machine	Destination after propagation
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
Remote Non-IHS Web Server Unmanaged Node	None	Files must be copied manually Recommended location: <Plug-in Install>/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

31

Web Server Definition and Plug-in

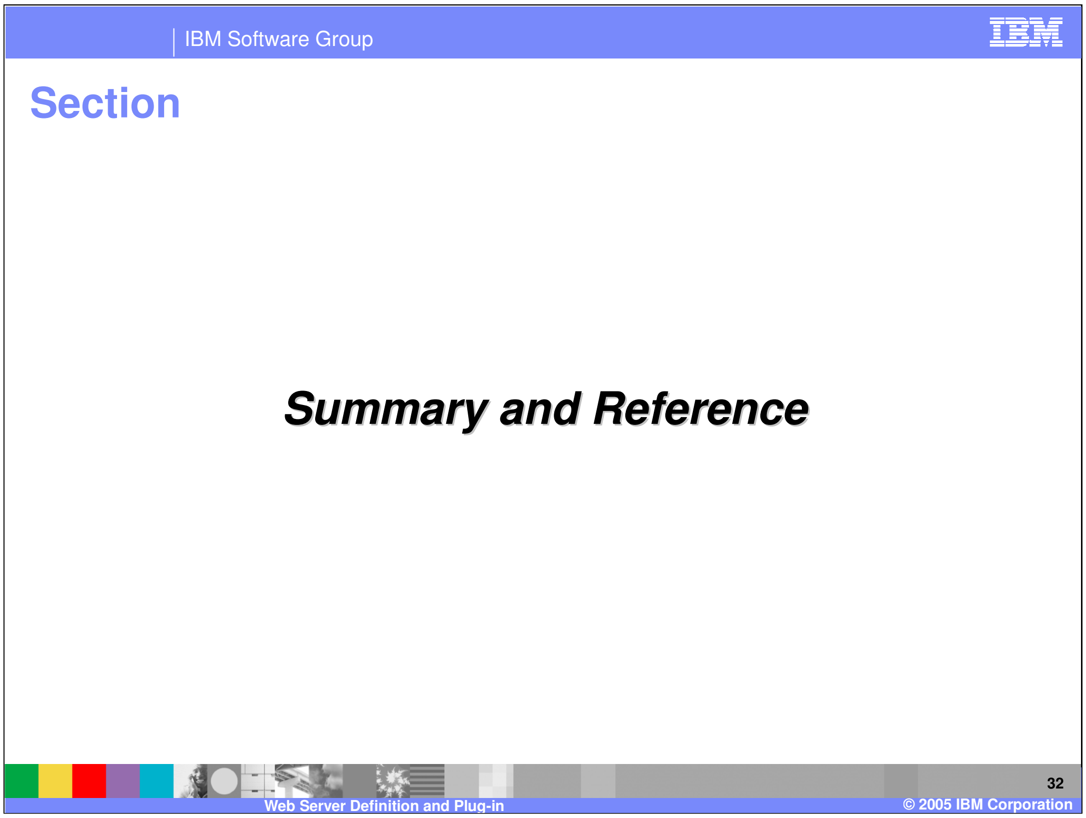
© 2005 IBM Corporation

This slide details the plug-in automatic propagation and destination location on a remote machine.

Local Web Server are configured directly with plug-in location directly where the plug-ins are generated. No propagation is necessary.

IHS Web Server configured as unmanaged node , the files are propagated under plug-in installation directory <Plug-in Install>/config/WebServerName/plugin-cfg.xml file

For Managed/Custom node the node sync operation takes care for replicating the configuration.



IBM Software Group

Section

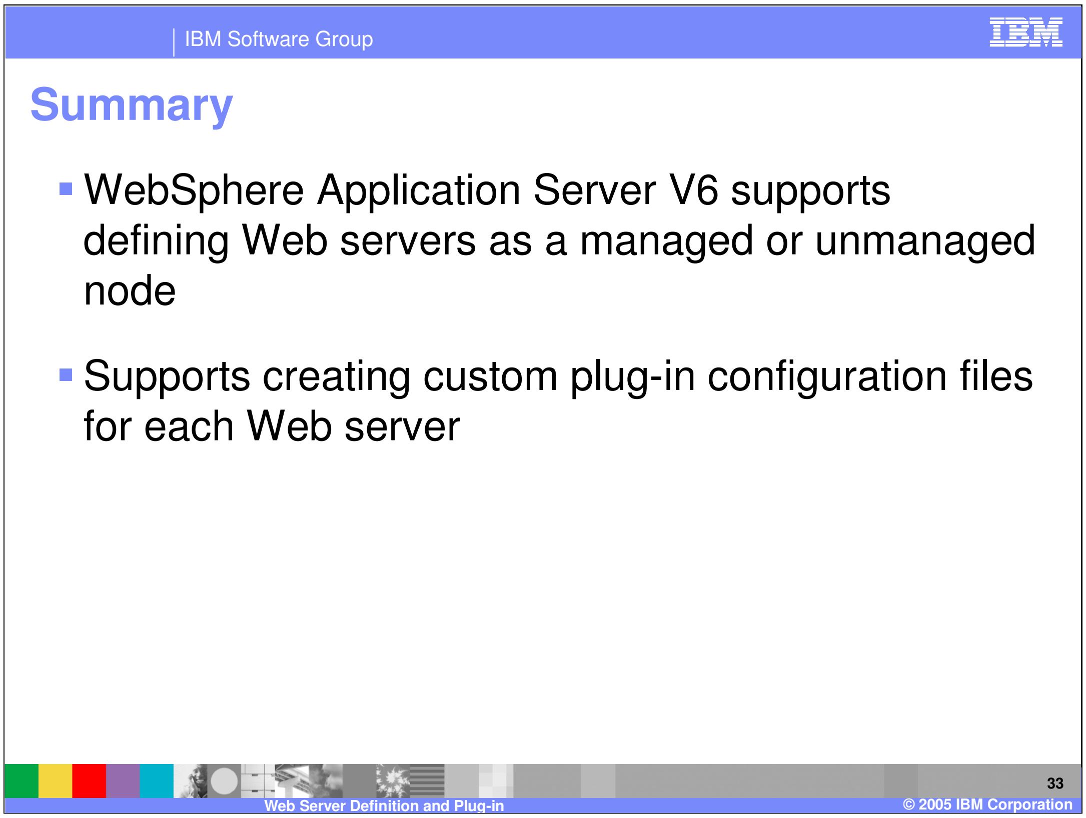
Summary and Reference

Web Server Definition and Plug-in

32

© 2005 IBM Corporation

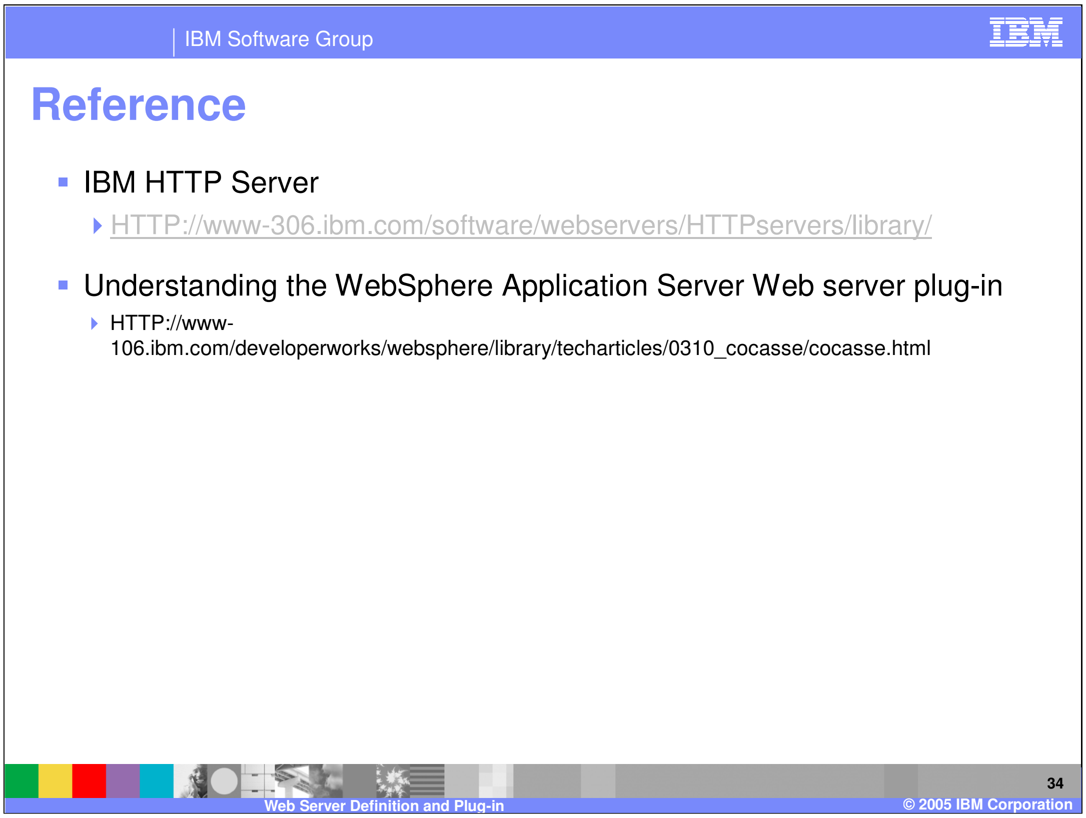
Next will be the summary and reference section.



Summary

- 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 for support for web servers. It detailed the concepts behind managed and unmanaged nodes for web servers. It also explained the special capabilities offered by the IBM HTTP Server V6.



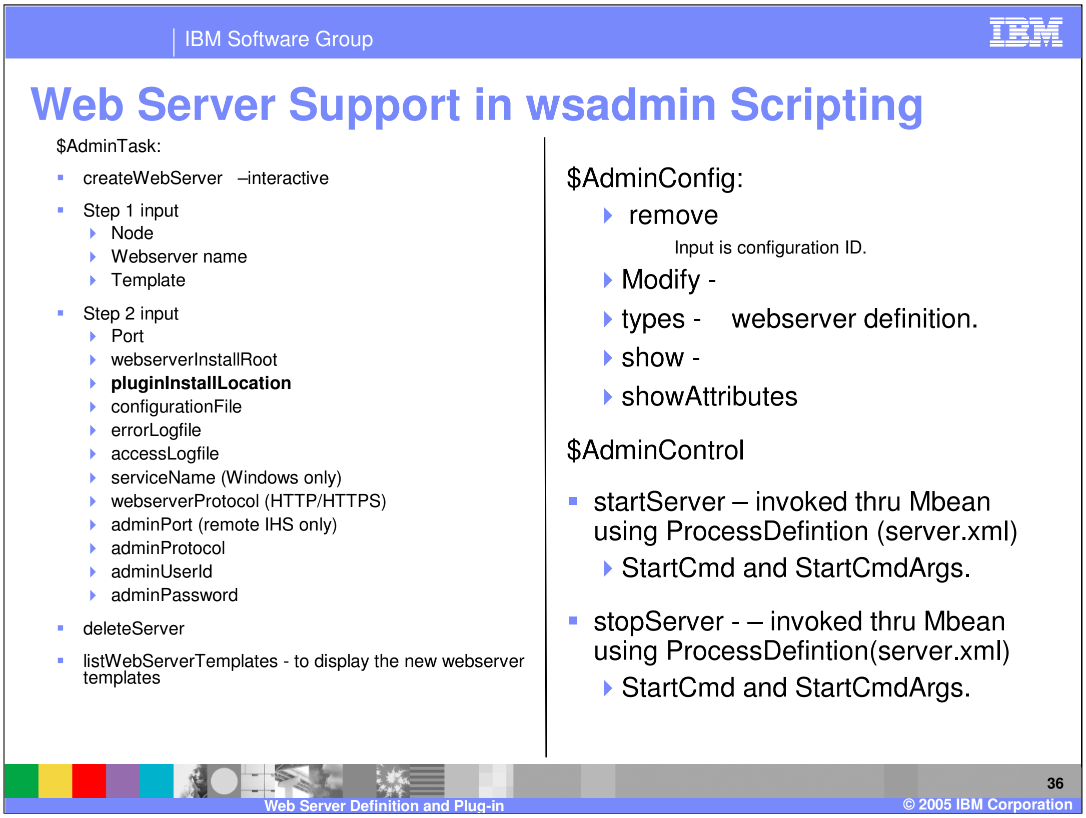
Reference

- 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

Appendix

Wsadmin commands to create Web server definition

Appendix shows the 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:

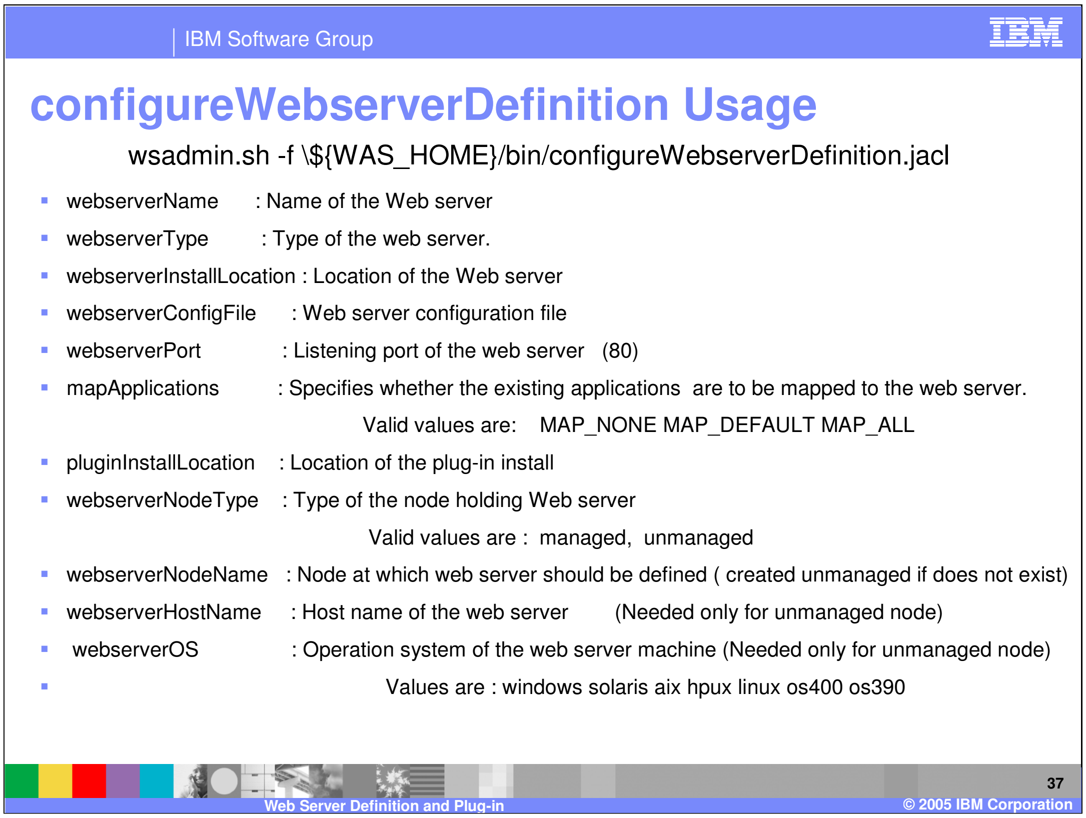
- remove

  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.



IBM Software Group

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

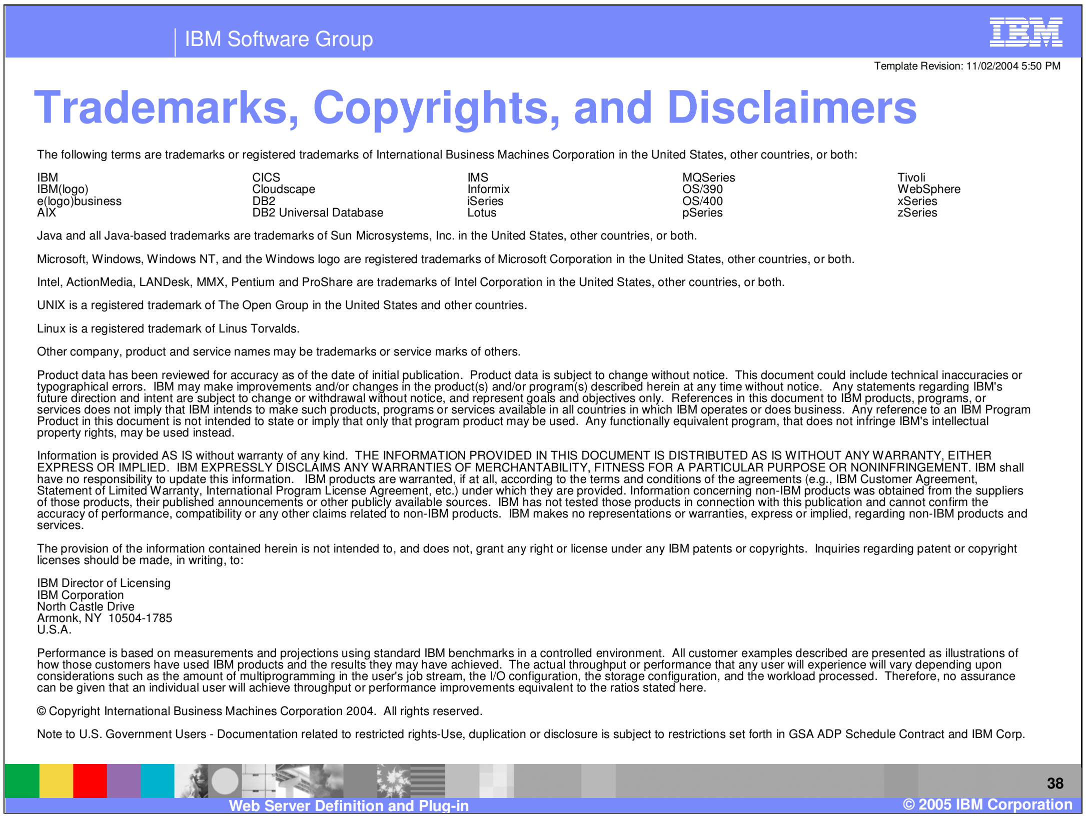
  Valid values are : managed, unmanaged
- webserverNodeName : Node at which web server should be defined (created unmanaged if does not exist)
- webserverHostName : Host name of the web server (Needed only for unmanaged node)
- webserverOS : Operation system of the web server machine (Needed only for unmanaged node)
- Values are : windows solaris aix hpux linux os400 os390

Web Server Definition and Plug-in

37

© 2005 IBM Corporation

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



Template Revision: 11/02/2004 5:50 PM

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

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.