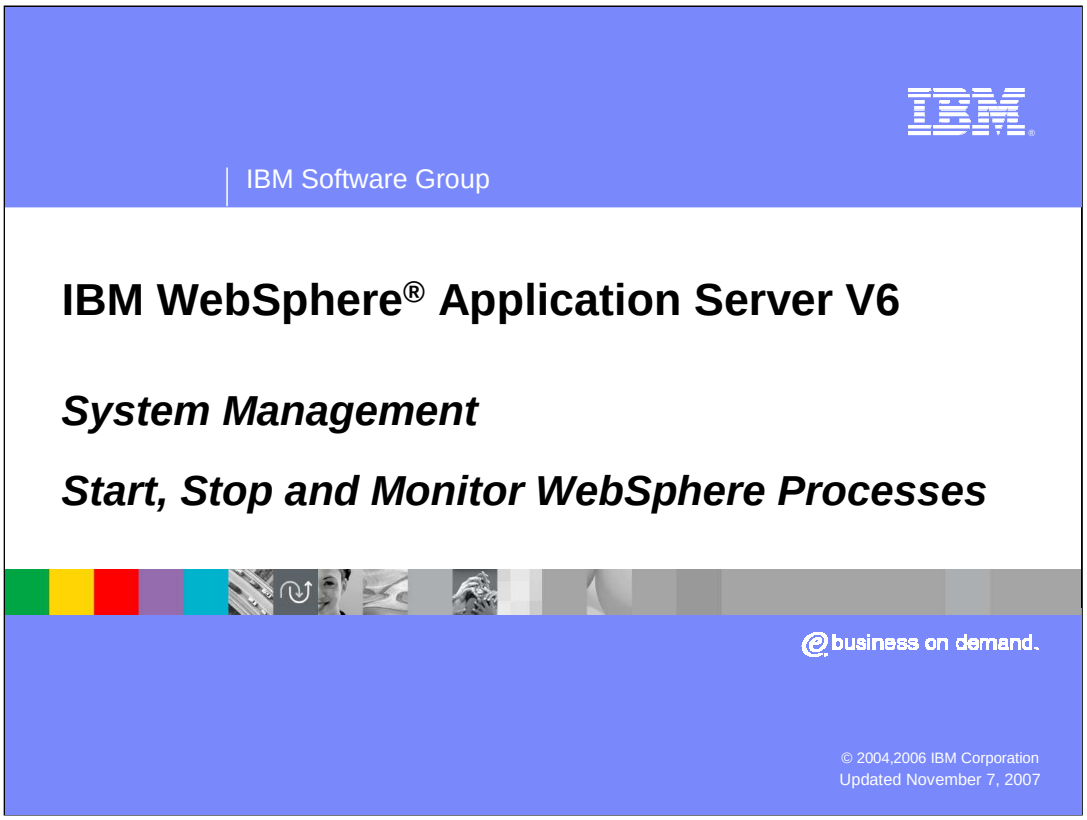



This presentation will cover stopping and starting WebSphere processes.

## Goals

- Describe how to start, stop and monitor WebSphere Application Server processes
- Pre-requisites:
  - Basic understanding of WebSphere Application Server V6 System Management Architecture




The goal of this presentation is to describe both Administrative Console and Command line alternatives for stopping the WebSphere processes.

# Agenda

- Start/Stop Application Server in Stand-alone Node
- Start/Stop Application Server processes in Network Deployment
- Monitoring Application Server processes to automatically start them




The agenda for this presentation is to explain the stopping and starting options for a stand-alone node, then for the processes in a cell, and finally setting up the operating system to automatically start WebSphere processes.

## Section

# *Starting and Stopping Stand-alone Application Server*




This section will cover starting and stopping the application server on the stand-alone node.

# Starting Application Server – Different options

- Command line
  - Run from the <PROFILE_HOME>\bin directory
    - Example: startServer server1
  - Run from <INSTALL_HOME>\bin directory specifying profile name
    - Example: startServer –profileName AppSrv01 server1
  - Logs in <PROFILE_HOME>\logs directory
- Windows® Start menu
  - Start → Programs → IBM WebSphere → Application Server V6.0 → Profiles → <Profile> → Start the Server
- If Server was registered as Windows Services
  - Registration using WASService.exe or at the time of creating the profile
  - Control Panel → Administrative Tools → Services




On a Windows machine, you can start the application server from the command line by entering startServer followed by the server name. If no profile name is given, the default profile is started. You can also start the application server from the Start menu as indicated. Finally, if you have used the WASService tool to register the application server as a Windows service, you can start the application server from the control panel.

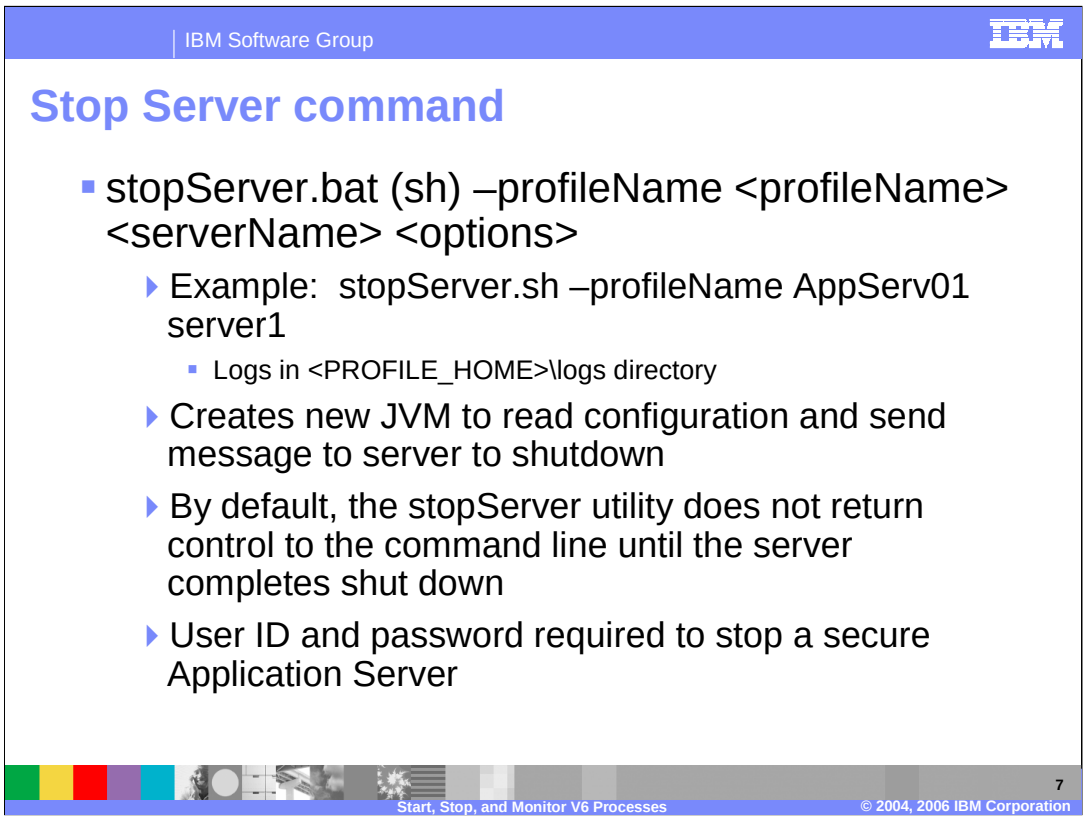
## Start Server command

- startServer.bat(sh) –profileName <profile> <serverName> <options>
  - Creates new Java™ Virtual Machine (JVM) to parse configuration
  - Launches Application Server specified by <serverName> in profile specified by <profileName> with JVM options specified in server.xml of that server
  - Refer to the V6 Information center for other options
- "-script" option creates a launch script for server1, does not start the server.
  - startServer server1 –script launchServer1.sh
  - Launch Scripts can be used to start the server (JVM)
  - Reduces start time by not parsing configuration files
  - If the JVM settings are changed, create a new launch script




This slide shows the command line tools available for starting and stopping servers in a stand-alone Application Server installation. The startServer.sh (or startServer.bat in Windows) takes as input the name of the server to be started. This command should be run from the bin directory of your WebSphere install. The startServer command causes a JVM to be started, which then reads the configuration information about the server from the configuration repository, and then launches the server. Any logs generated from the command go to the startServer.log file in the logs directory for the server being started. The process ID of the server that was started is returned from the startServer command.

The –Script option creates a startup script or batch file that will hard code the configuration into the startup script, and thereby reduce startup times by shortcutting the processing of the configuration files.




stopServer is similar to startServer. However, it sends a message to the running server to tell it to stop. Unless invoked with the “nowait” option, it will not return until the server is fully stopped.

Full documentation of these commands along with all the available options is provided in the Information Center.

## Section

# *Starting and Stopping WebSphere Application Server Processes for a Network Deployment Cell*

# *(Application Servers, Node Agent, DMgr)*




This section will discuss starting and stopping application server processes in the ND environment.

## Starting Application Server

- Using startServer command: Same as Stand-alone Application Server
  - Command: startServer –profile <profileName> <serverName> options
- Node Agent must be running before starting the Application Server
- Can also stop from the Deployment Manager (DMgr) Administrative Console or using wsadmin command




The startServer command is used exactly the way it is for a stand-alone node. Except in limited circumstances, the Node Agent should be running before the application server is started.

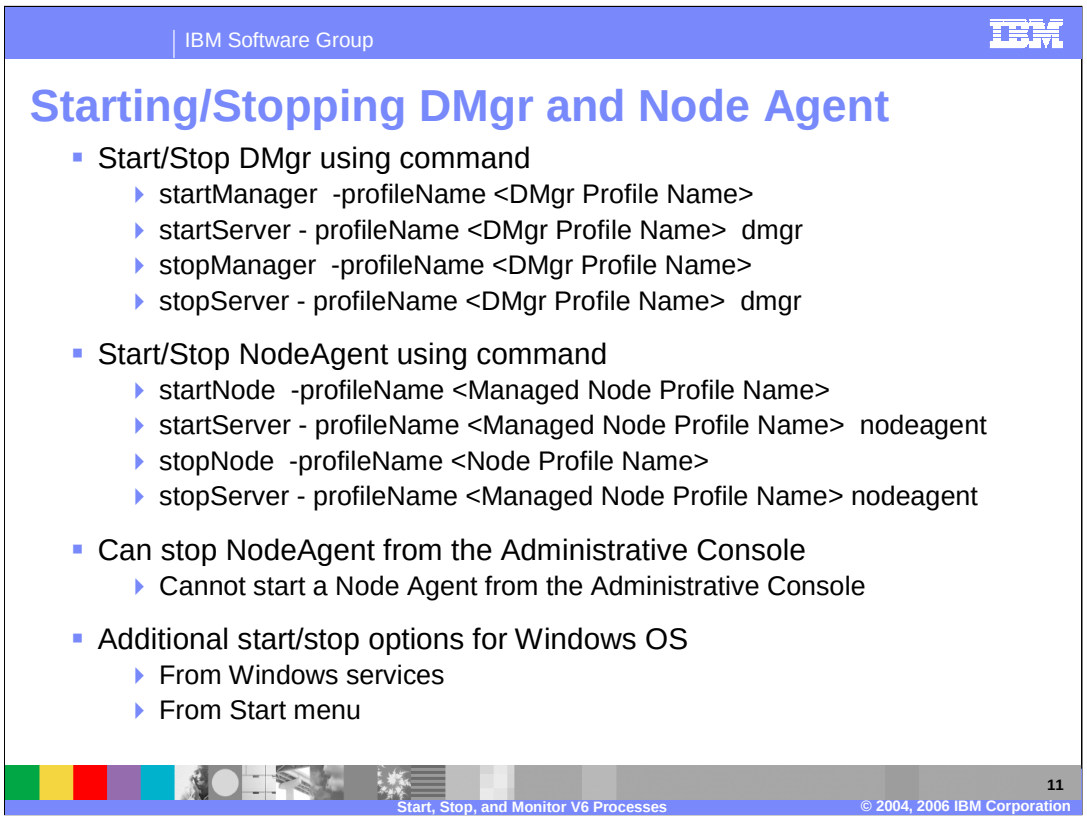
In a Network Deployment cell, the Administrative Console has a Start button on the Servers list. Navigate to Application Servers, and select the checkbox in front of the server name, and click Start. To launch an application server from wsadmin, use (JACL) $AdminControl startServer serverName nodeName, or (Jython) AdminControl.startServer('serverName', 'nodeName').

## Stop Server command

- Using stopServer command: Same as Stand-alone Application Server
  - stopServer.bat (sh) –profile <profileName> <serverName> options
- Can also stop from the Administrative Console or using wsadmin command




The command line syntax for stopServer is the same as for startServer. If security is enabled, you will have to include a –user and a –password parameter to stop a server.




In the stand-alone application server environment, you stop and start servers from the command line.

In the Network Deployment environment, you can stop and start Application Servers and JMS servers using the Administrative Console. However, the node agents and the deployment manager are still started from the command line, or through wsadmin.

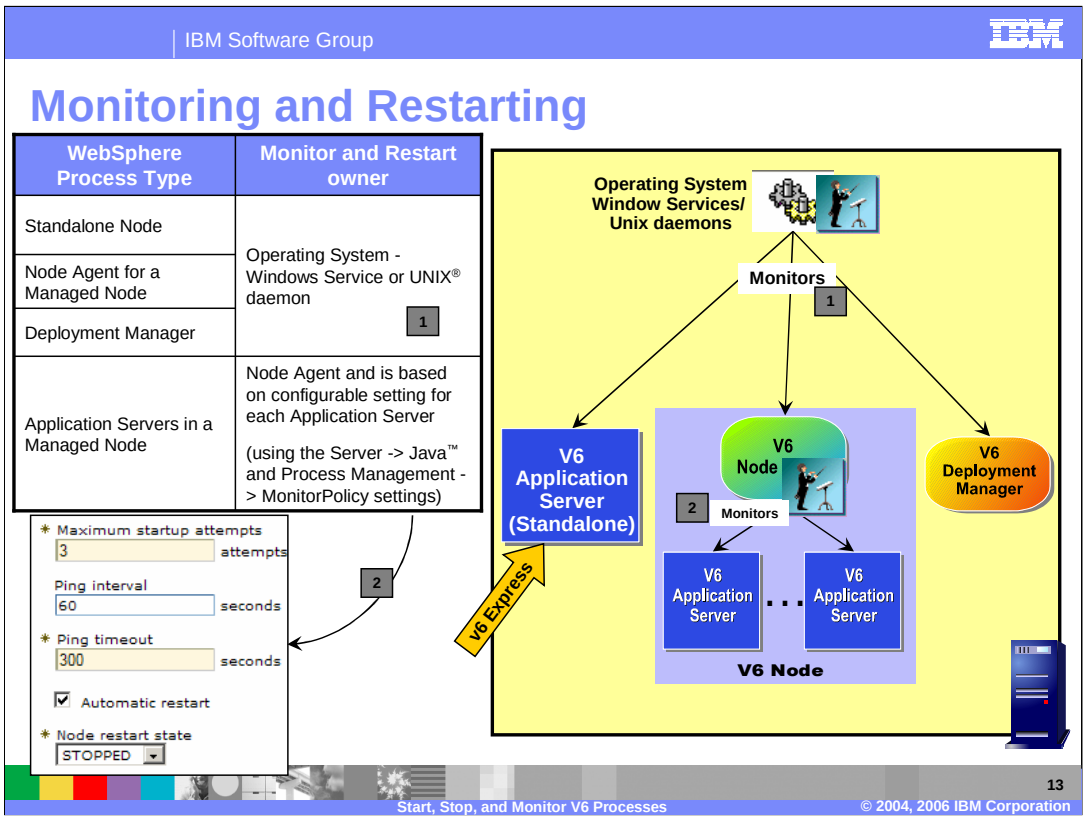
In the z/OS® environment, it is more likely that the MVS start and stop task commands will be used. However, the WebSphere provided command line tools and the Administrative Console are still viable options on z/OS.

## Section

# *Monitoring WebSphere Application Server Processes*



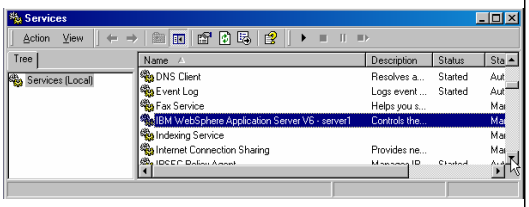

This section will cover monitoring WebSphere processes.



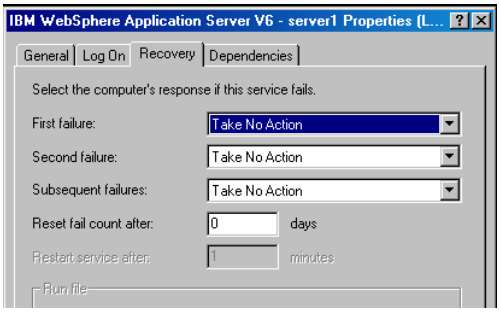

Operating system should be configured to monitor the stand-alone Application Server, Node Agent and deployment manager servers in case of failure or to automatically start during a reboot. In a Network Deployment environment, the **addNode** or **startNode** command starts a single unmonitored node agent only, the node agent process, and does not start all of the processes that you might define on the node. While running, the node agent monitors and restarts Application Server processes on that node, on either a Windows or a Linux® and UNIX-based platform. Each Application Server process has MonitoringPolicy configuration settings that the node agent uses when monitoring and restarting the process.

# Windows OS: Monitoring Server Process

- Windows services
  - Optionally, created during defining new Profile for Stand-alone Node, Managed Node or DMgr
  - addNode command has option to add Node Agent services
  - WASService.exe command to insert service registry entry




- User must belong to Administrator group and have the user rights for
  - "*Act as part of the operating system*"
  - "*Log on as a service*"
- Can configure the services to:
  - Automatic/Manual start
  - Recovery actions







There are several server processes related to WebSphere Application Server products that the operating system can monitor and automatically restart when the server processes stop abnormally.

You can create Windows services during installation, using the installation wizard. You can use the **WASService** command in the *install_root*/bin directory to do so at a later time. Refer to the V6 Information Center for more information.

You can configure a base Application Server as a WebSphere Application Server monitored process.

## WASService.exe (Windows only)

- WASService command line tool allows adding any WebSphere Application Server process as a Windows service, and managing the service
- WASService.exe command syntax for adding a service
  - WASService.exe -add <*service_name*> -serverName <*Server*> –profilePath <server's profile path> <options>
- WASService.exe command syntax for managing services
  - WASService.exe -remove *service_name*
  - WASService.exe -start *service_name*
  - WASService.exe -stop *service_name*
  - WASService.exe -status *service_name*
- For more details on WASService, refer to the V6 Information Center




On a Windows machine, the WASService command allows you to register a Windows service so that the operating system can monitor and if needed, restart a WebSphere process.

# UNIX: Monitoring Server Process

UNIX Machine

etc/inittab

**rc.was**

numRetries=3

launchScript=start_server1.sh

- Sample rc.was shell script can be found in <WAS_ROOT>/bin directory
  - Starts a defined process using a launch script
    - Launch script can be created using startServer command with –script option
    - Example: StartServer server1 –script start_server1.sh
  - Restart process for non-zero exit
    - Number of retries can be customized (numRetries=3)
- System administrator has to manually add an entry into inittab
  - *was:235:once:/usr/WebSphere/AppServer/bin/**rc.was** >/dev/console 2>&1*
    - Check to see what runlevel your machine is booting by looking in the inittab file - make sure you specify the correct runlevel in your new entry – the example above specifies a run level of 2,3 and 5
    - Explore the use of "***boot***" or "***respawn***" options in your "inittab" entry, if the option "*once*" does not work properly for you
- Servers started using startServer command will not be monitored





To set up this function on a Linux and UNIX-based operating system, you must have root authority to edit the inittab. There is a sample script in the bin directory that can be used as a guide for setting up your own scripts.

This script is intended to be used in the inittab file as a monitor for a WebSphere process.

## UNIX: Monitoring server process : rc.was Example

### rc.was Example

```
launchScript=start_server1.sh
numRetries=3 ; RETRY=0
while [ $RETRY -lt $numRetries ]
      do
            $binDir/$launchScript      # launching server using $launchScript
             rc=$?
             echo exit code: $rc
            # Increment retry count on anything other than a normal exit code
             if [ $rc -gt 0 ]
             then
                  RETRY=`expr $RETRY + 1`
             fi
            case $rc in
                  0) break ;;
```

- Typically Stand-alone Application Servers, Node Agent and Deployment Manager are monitored
- Multiple scripts and entries in inittab, may be required




This slide illustrates the sample rc.was script

## Summary

- Stand-alone Application Server can be started and stopped locally
- The Deployment Manager can start, stop and monitor processes in a Network Deployment cell




In summary, this presentation has described methods for stopping and starting WebSphere processes.

# Trademarks, Copyrights, and Disclaimers

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

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

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

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

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

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

Linux is a registered trademark of Linus Torvalds.

Other company, product and service names may be trademarks or service marks of others.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements and/or changes in the product(s) and/or program(s) described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (e.g., IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.



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.