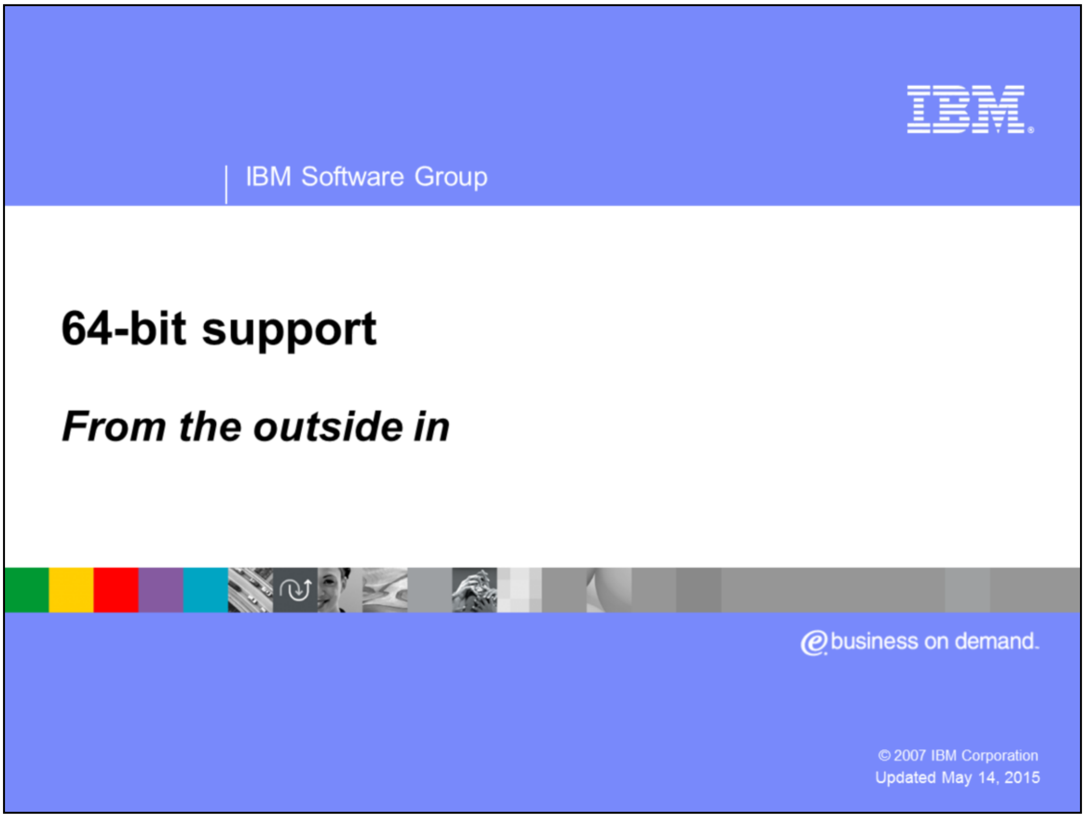


IBM Software Group

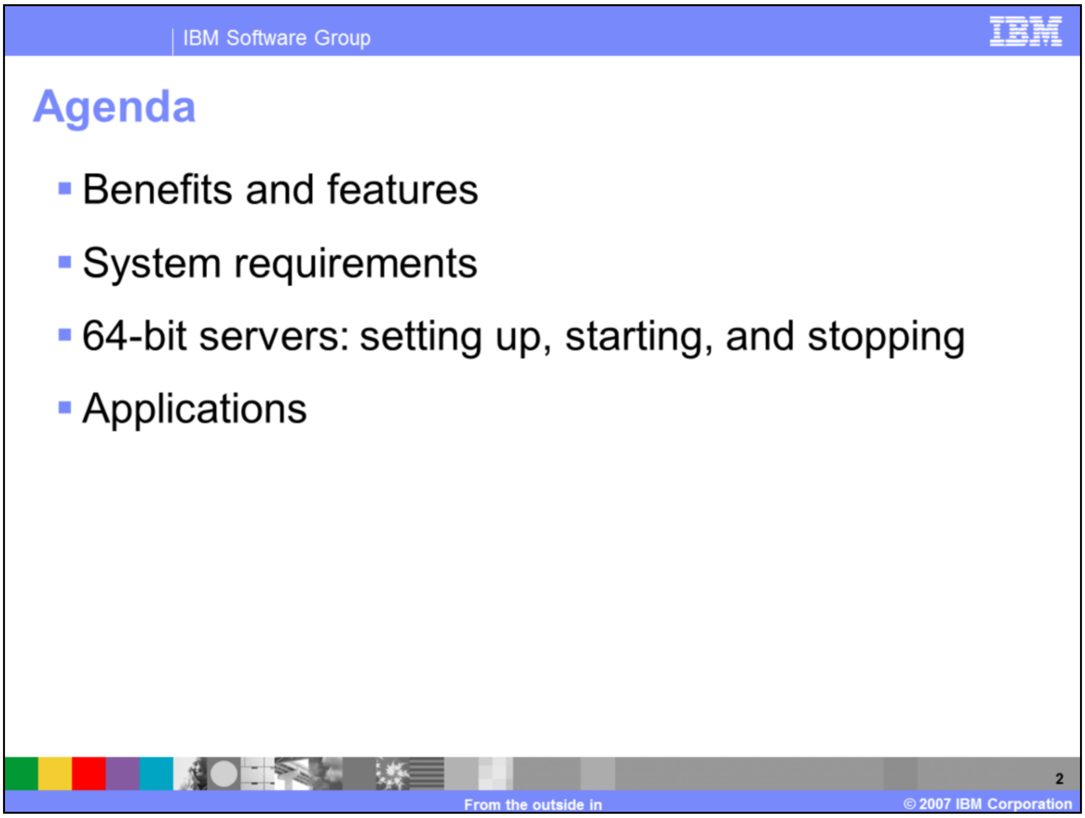
64-bit support

From the outside in

business on demand.

© 2007 IBM Corporation
Updated May 14, 2015

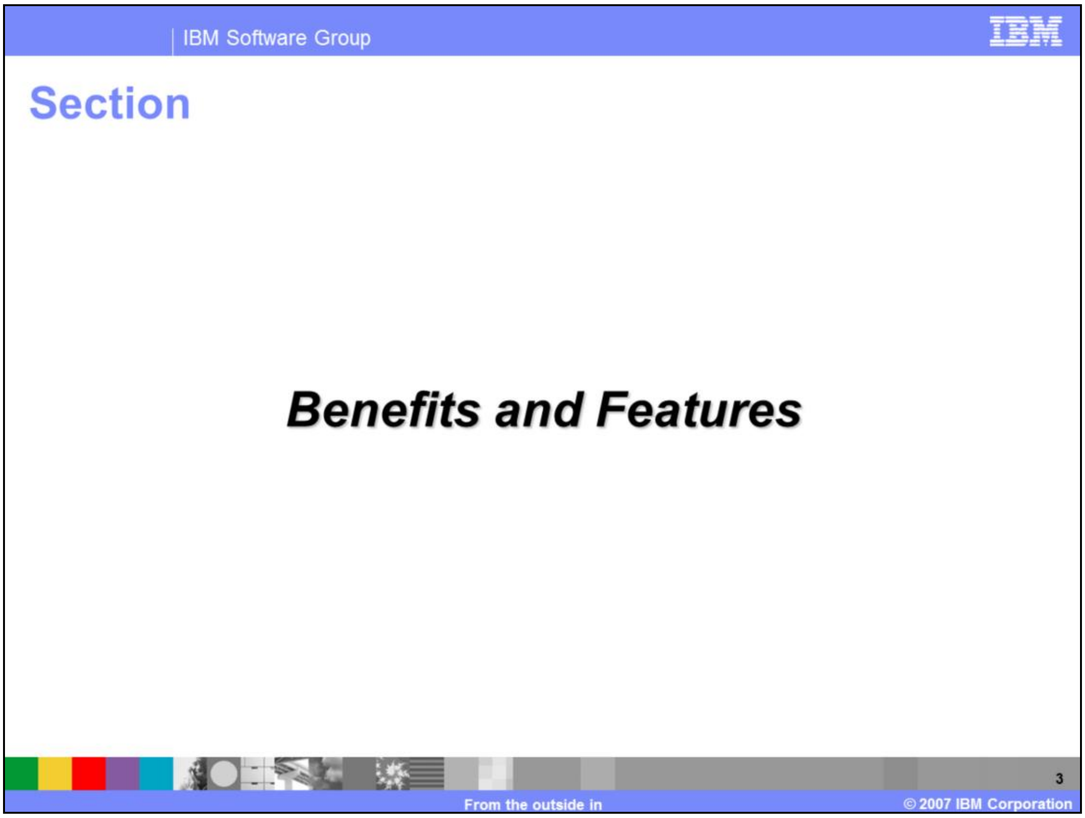
This presentation will show an overview of 64-bit support in a WebSphere Base Application Server V6.1 on z/OS.



Agenda

- Benefits and features
- System requirements
- 64-bit servers: setting up, starting, and stopping
- Applications

This presentation will introduce the benefits and features of 64-bit processing in a WebSphere Application Server Version 6.1 system structure on z/OS. System requirements, 64-bit server processing and applications will also be discussed.



IBM Software Group

Section

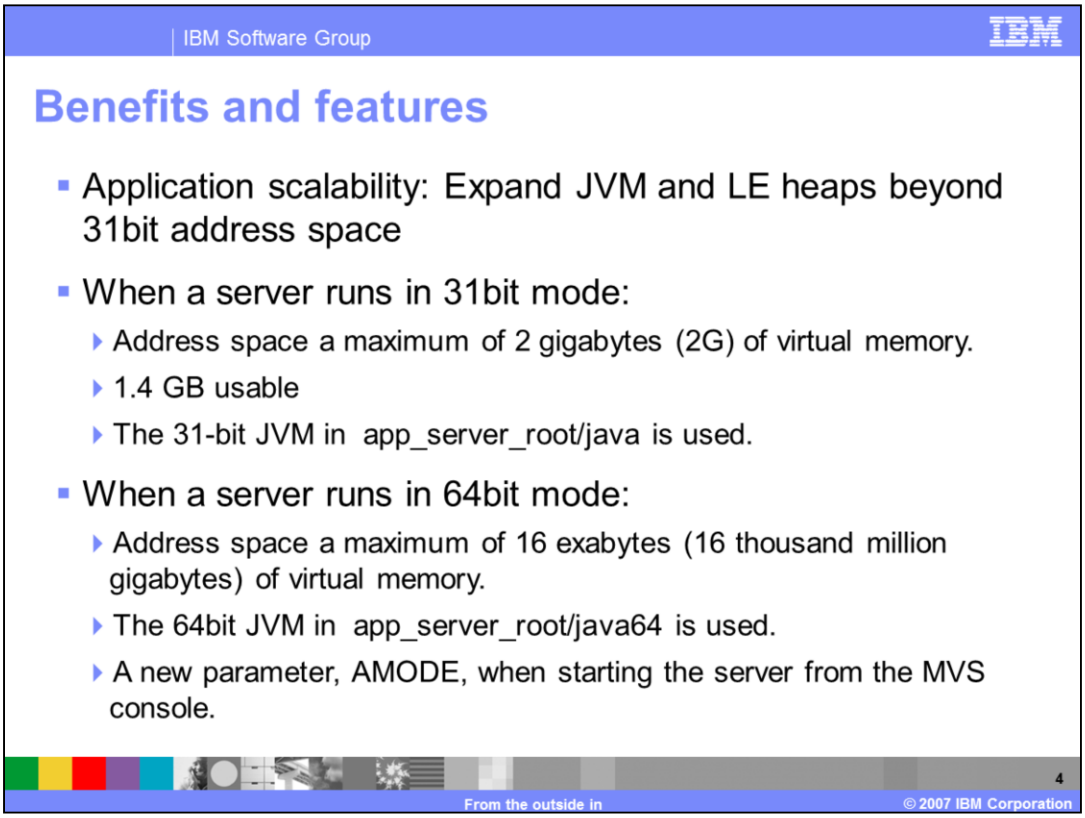
Benefits and Features

3

From the outside in

© 2007 IBM Corporation

This section will discuss the Benefits and Features of 64-bit processing on z/OS.



IBM Software Group

Benefits and features

- Application scalability: Expand JVM and LE heaps beyond 31bit address space
- When a server runs in 31bit mode:
  - Address space a maximum of 2 gigabytes (2G) of virtual memory.
  - 1.4 GB usable
  - The 31-bit JVM in app_server_root/java is used.
- When a server runs in 64bit mode:
  - Address space a maximum of 16 exabytes (16 thousand million gigabytes) of virtual memory.
  - The 64bit JVM in app_server_root/java64 is used.
  - A new parameter, AMODE, when starting the server from the MVS console.

4

From the outside in

© 2007 IBM Corporation

64-bit support answers a requirement for greater application growth.

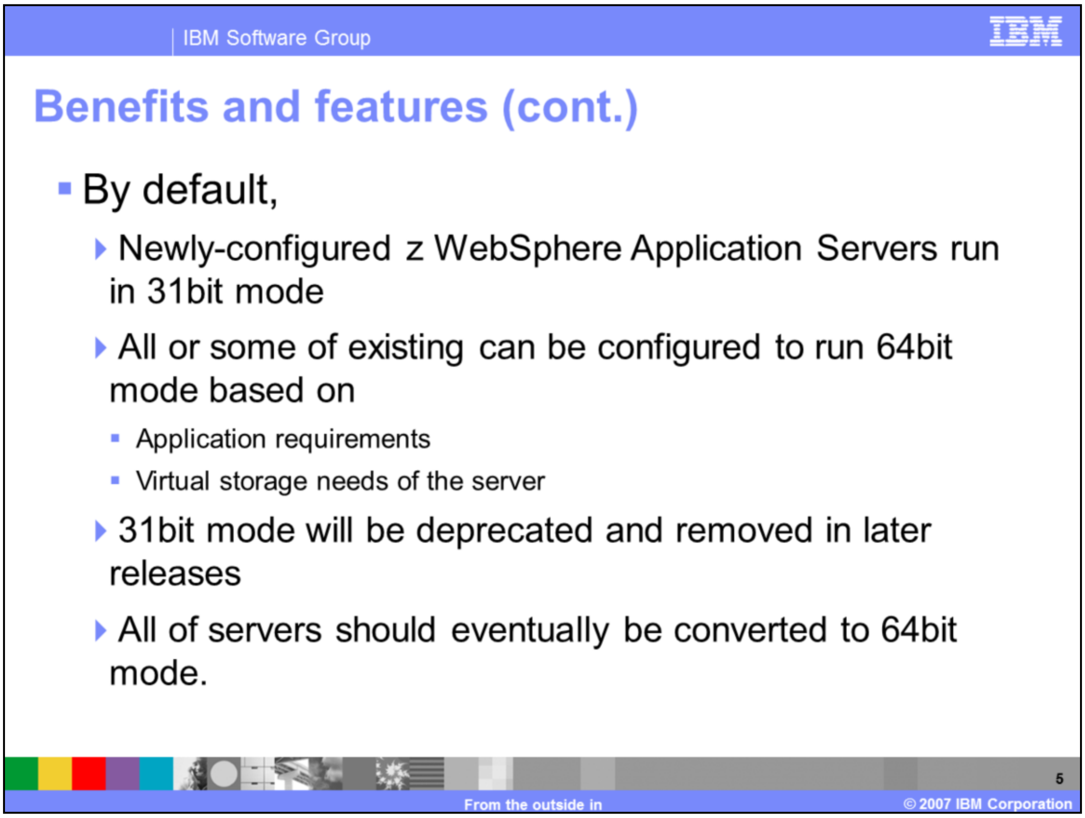
When a server runs in 31-bit mode each server address space can access a maximum of 2 gigabytes of virtual memory. When you account for virtual memory used by the operating system and other subsystems, the amount left over is approximately 1.4 gigabytes. The 31-bit JVM defined in the app_server_root/java variable is used.

In 64-bit mode the heaps for the Java™ Virtual Machine and the C language Environment can move beyond the 2 gigabyte address limitation of 31-bit addressing. The phrase “above or below the bar” is used when talking about addresses above or below 2 gigabytes. In the past, storage above or below “the line” was used for 24-bit verses 31-bit addressing; with 64-bit, it’s above or below “the bar”.

In 64-bit mode, each server address space can access a maximum of 16 exabytes, 16 thousand million gigabytes, of virtual memory.

The 64-bit JVM defined in the app_server_root/java64 variable is used. A special parameter, AMODE, must be specified when starting the server from the MVS console.

Also, garbage collection can occur less frequently in larger heaps as the collection threshold gets hit less often.



IBM Software Group

Benefits and features (cont.)

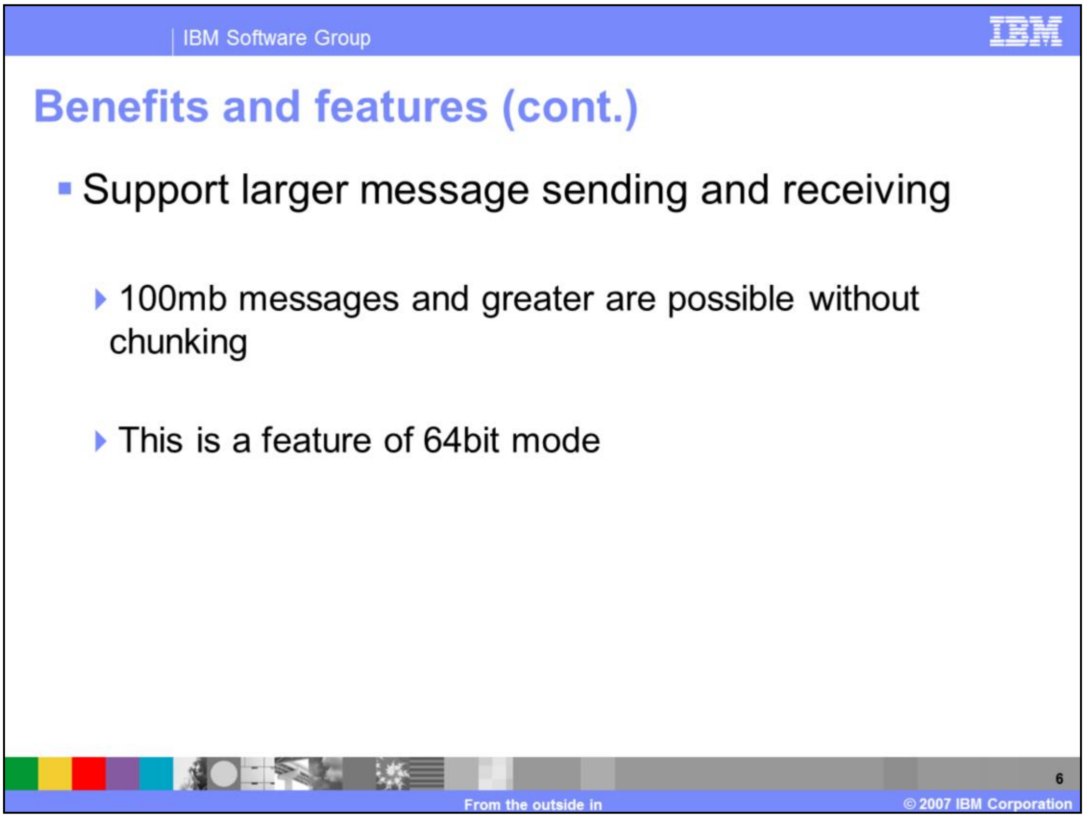
- By default,
  - Newly-configured z WebSphere Application Servers run in 31bit mode
  - All or some of existing can be configured to run 64bit mode based on
    - Application requirements
    - Virtual storage needs of the server
  - 31bit mode will be deprecated and removed in later releases
  - All of servers should eventually be converted to 64bit mode.

5

From the outside in

© 2007 IBM Corporation

By default, newly-configured WebSphere for z/OS servers run in 31-bit mode. A customer can convert some or all of their servers to run in 64-bit mode, based on the application requirements and virtual storage needs of each server.   Servers do not need to be converted to 64-bit mode immediately, however, since 31-bit mode will be deprecated and removed in later releases, conversion to 64-bit mode will be required eventually.



IBM Software Group

Benefits and features (cont.)

- Support larger message sending and receiving
  - 100mb messages and greater are possible without chunking
  - This is a feature of 64bit mode

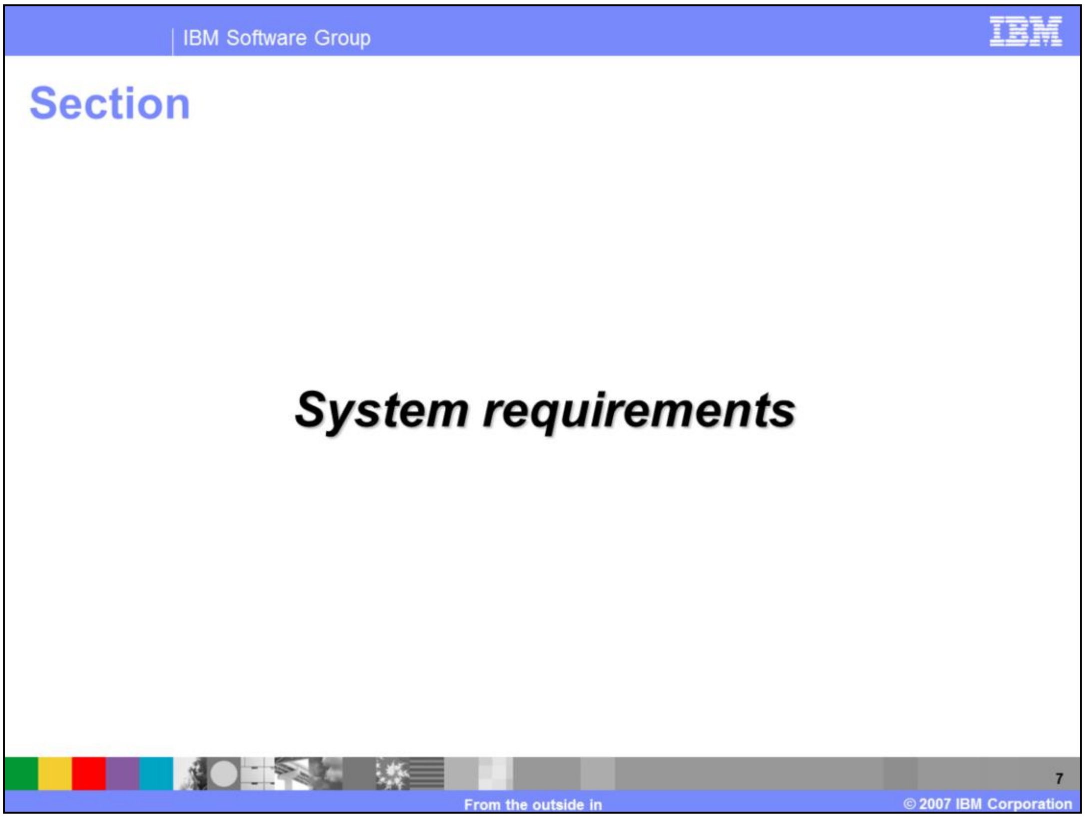
6

From the outside in

© 2007 IBM Corporation

A functional feature benefit of 64-bit mode is support for larger sizes of sent messages and responses. Messages of varying sizes, beyond the current 10 megabyte limit, can send and receive up to and beyond 100 megabytes. Greater sizes are possible but that will depend on the individual installation because different system capacities or constraints will affect how well a given server will handle larger messages.

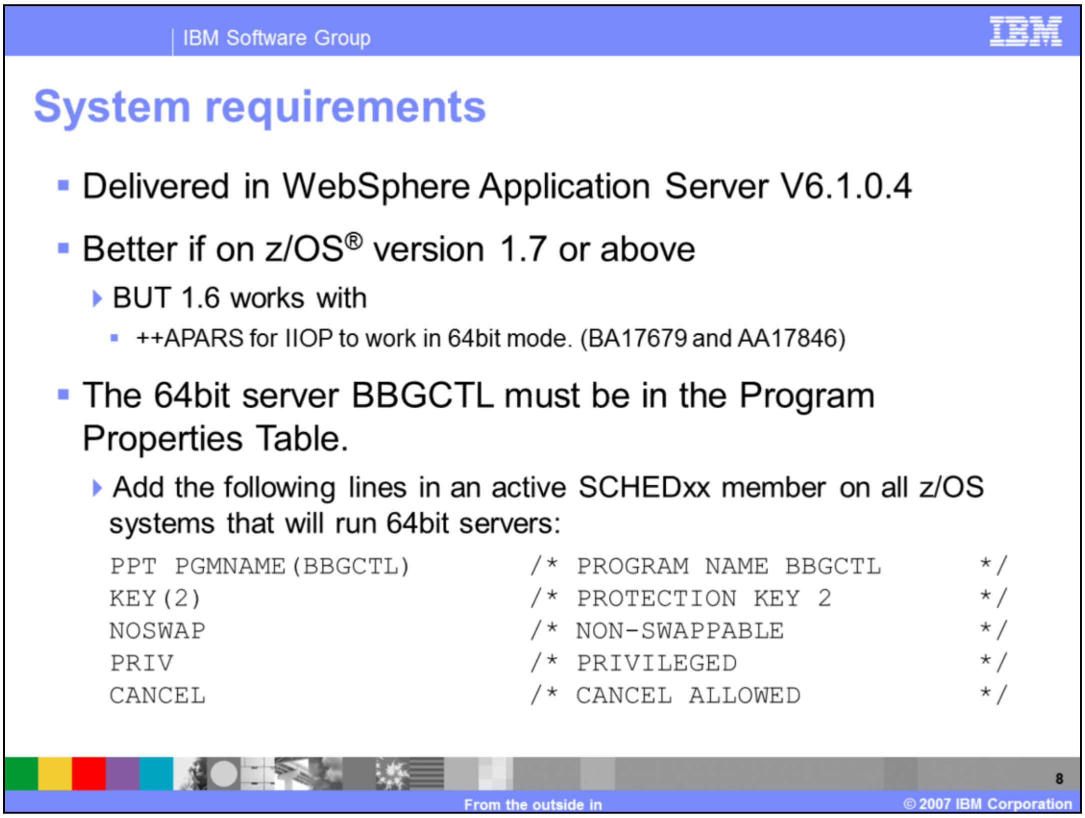
In 31-bit mode the 10 megabyte limit still exists.



Section

System requirements

This section will discuss System Requirements of 64-bit processing on z/OS.



IBM Software Group

System requirements

- Delivered in WebSphere Application Server V6.1.0.4
- Better if on z/OS® version 1.7 or above
  - BUT 1.6 works with
    - ++APARS for IIOP to work in 64bit mode. (BA17679 and AA17846)
- The 64bit server BBGCTL must be in the Program Properties Table.
  - Add the following lines in an active SCHEDxx member on all z/OS systems that will run 64bit servers:

```
PPT PGMNAME(BBGCTL)          /* PROGRAM NAME BBGCTL       */
KEY(2)                       /* PROTECTION KEY 2          */
NOSWAP                       /* NON-SWAPPABLE             */
PRIV                         /* PRIVILEGED                */
CANCEL                       /* CANCEL ALLOWED            */
```

8

From the outside in

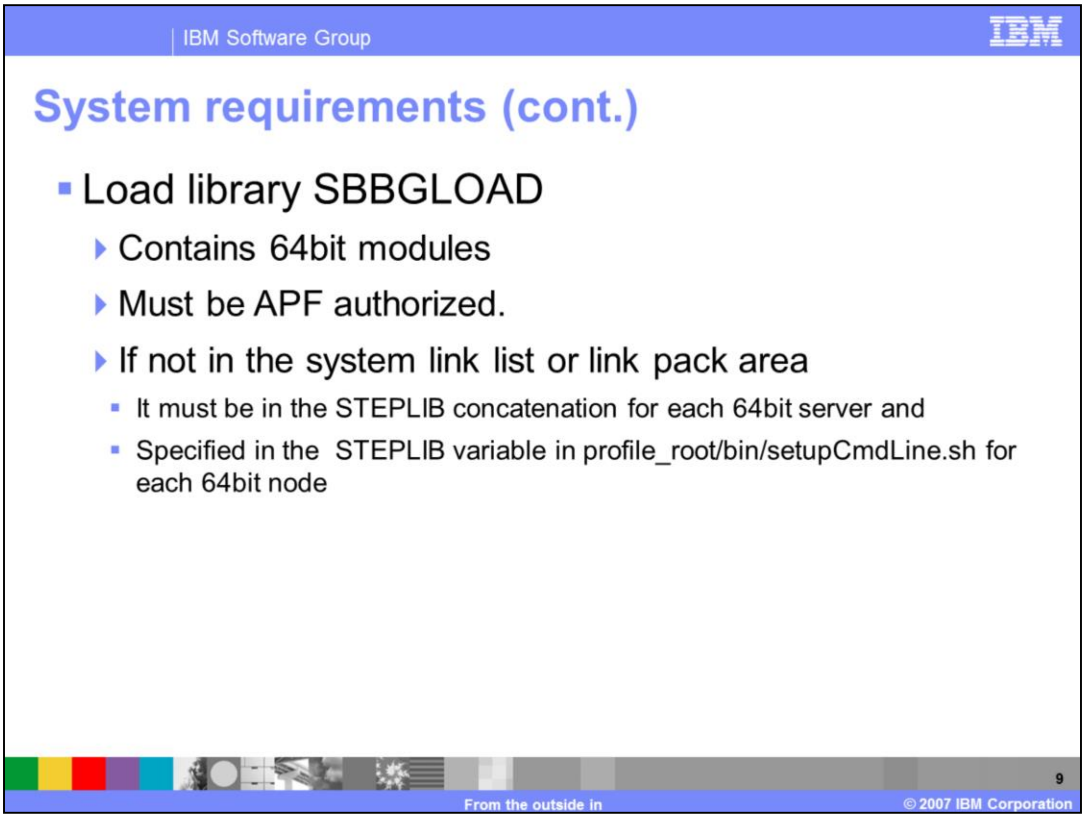
© 2007 IBM Corporation

So what is needed to run z/Websphere in 64bit mode?

First, any server running in 64-bit mode must be at WebSphere Application Server for z/OS Version 6.1.0.4 or above.

Second, z/OS level 1.7 or above is necessary to start the servers in 64-bit mode. There is a dependency on z/OS function regarding async I/O that is available in 1.7 otherwise IIOP connections will not work in 64-bit mode. This required function has been rolled back into 1.6 and is available from two ++APARS. If running z/OS 1.6, these fixes (BA17679 and AA17846) are necessary to install to run 64bit.

Finally, the 64-bit server module BBGCTL must be listed in the Program Properties Table. It is necessary for the lines shown here to be in an active SCHEDxx member on all z/OS systems that will run 64bit servers.



IBM Software Group

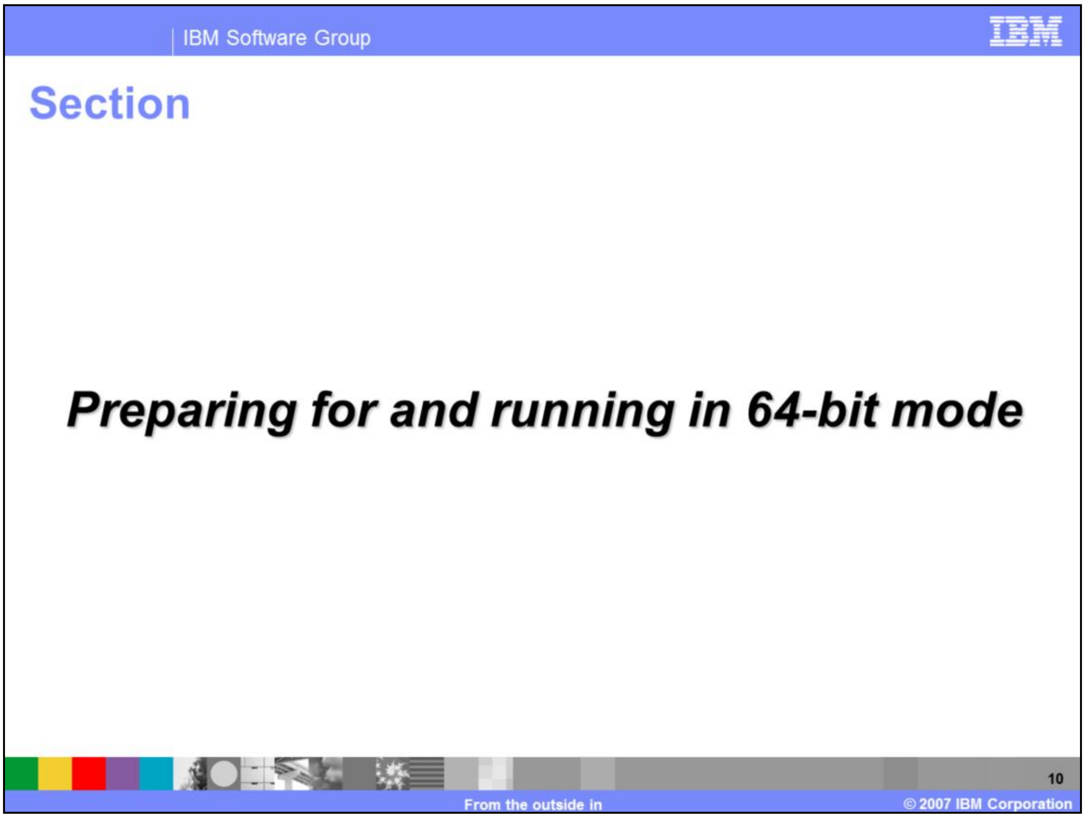
System requirements (cont.)

- Load library SBBGLOAD
  - Contains 64bit modules
  - Must be APF authorized.
  - If not in the system link list or link pack area
    - It must be in the STEPLIB concatenation for each 64bit server and
    - Specified in the STEPLIB variable in profile_root/bin/setupCmdLine.sh for each 64bit node

9

From the outside in

Product data set SBBGLOAD, which contains 64-bit support modules, must be APF authorized. If the SBBGLOAD data set is not in the system link list or link pack area, then it must be in the cataloged procedure STEPLIB concatenation for each server which is to run in 64-bit mode. Also, SBBGLOAD must be specified in the STEPLIB variable in profile_root/bin/setupCmdLine.sh for each node containing a server running in 64-bit mode.



IBM Software Group

Section

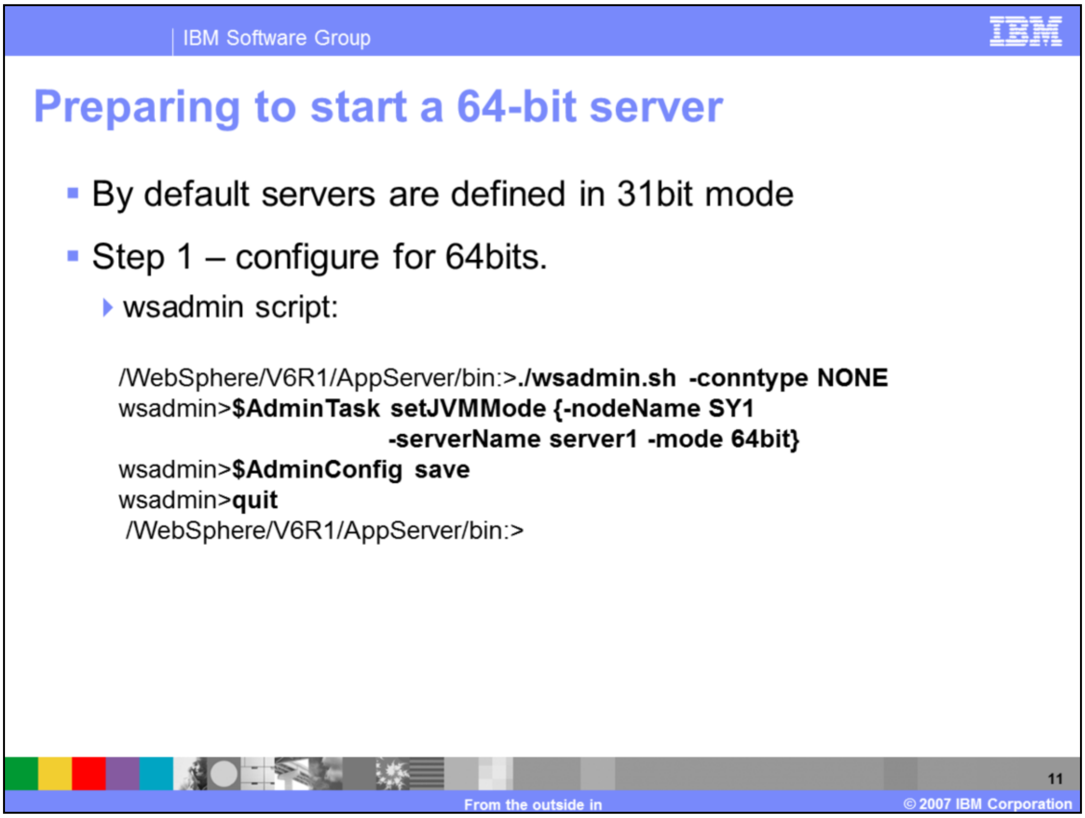
Preparing for and running in 64-bit mode

10

From the outside in

© 2007 IBM Corporation

This section will discuss preparation and running in 64-bit mode on z/OS.



IBM Software Group

Preparing to start a 64-bit server

- By default servers are defined in 31bit mode
- Step 1 – configure for 64bits.
  - wsadmin script:

```
/WebSphere/V6R1/AppServer/bin:>./wsadmin.sh -conntype NONE
wsadmin>$AdminTask setJVMMode {-nodeName SY1
                               -serverName server1 -mode 64bit}
wsadmin>$AdminConfig save
wsadmin>quit
 /WebSphere/V6R1/AppServer/bin:>
```

11

From the outside in

© 2007 IBM Corporation

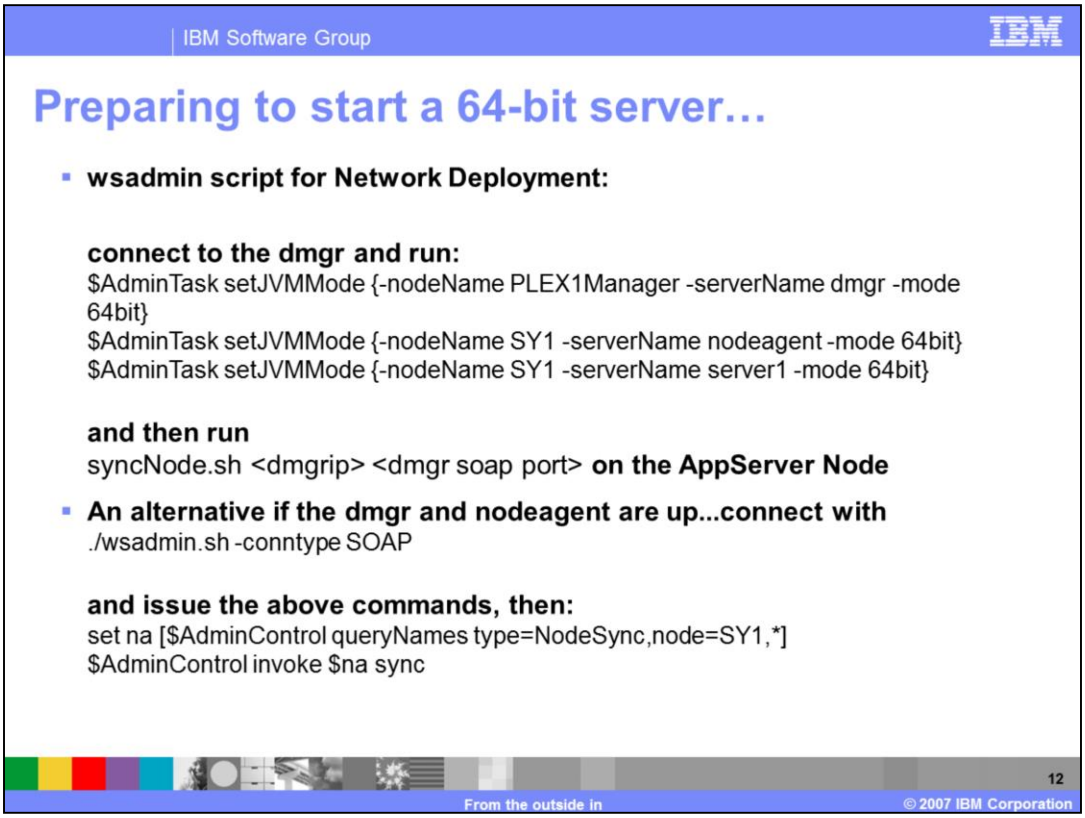
Preparing to run a server in 64-bit mode requires certain steps.

There is a necessary preparation step to configure the server to enable starting it in 64-bit mode. This is because by default, servers are defined for 31bit mode only. Certainly, existing servers are defined for 31bit mode. Any new servers that get defined, even after installing 64bit support, will be configured for 31bit mode. So a server needs to be specifically reconfigured to run in 64bit mode and there are a couple of ways to do that.

Let’s first consider using the wsadmin script. Once invoked, there is a new set of parameters. Use AdminTask setJVMMode with a mode of 64bit to configure the environment paths to point to the 64bit JVM and to update configuration files for 64bit mode.

AdminConfig save, commits the changes allowing the server to start in 64bit mode.

Note: if you want to UNconfigure 64bit mode from the server, you follow the same process as above but replace the –mode 64bit with –mode 31bit.



Preparing to start a 64-bit server…

- **wsadmin script for Network Deployment:**

  connect to the dmgr and run:

  ```
  $AdminTask setJVMMode {-nodeName PLEX1Manager -serverName dmgr -mode 64bit}
  $AdminTask setJVMMode {-nodeName SY1 -serverName nodeagent -mode 64bit}
  $AdminTask setJVMMode {-nodeName SY1 -serverName server1 -mode 64bit}
  ```

  and then run

  syncNode.sh <dmgrip> <dmgr soap port> **on the AppServer Node**

- **An alternative if the dmgr and nodeagent are up...connect with**

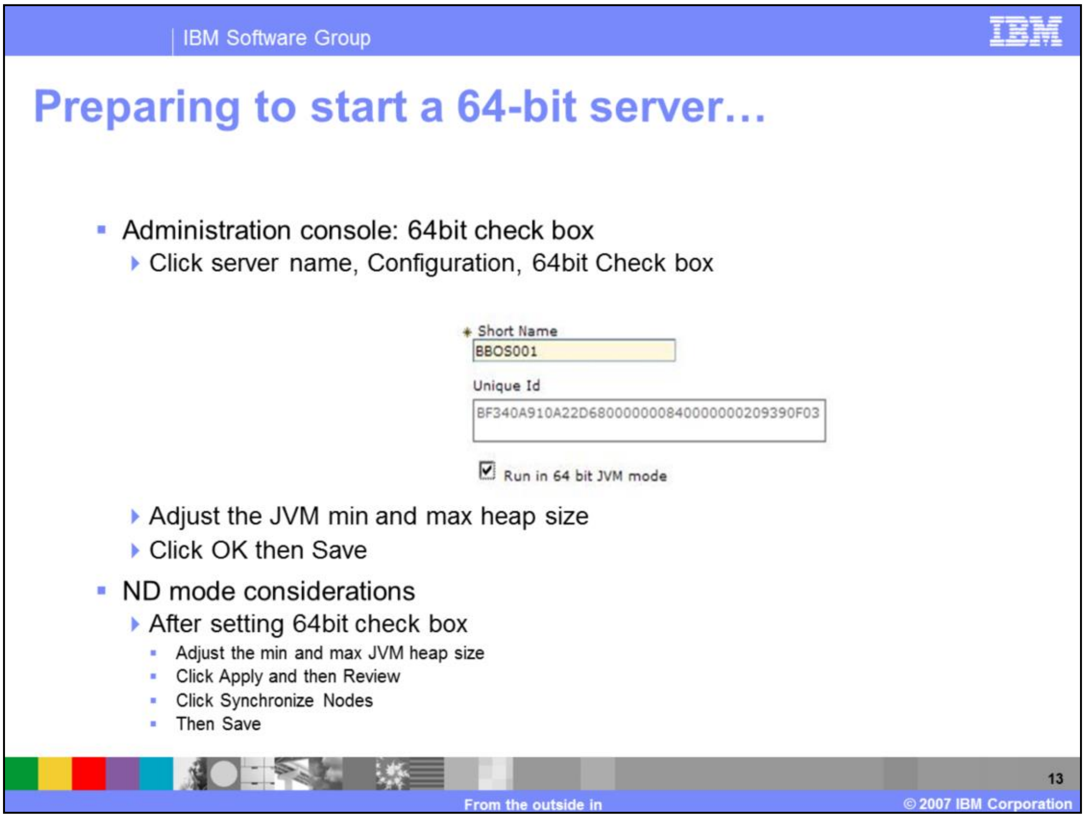
  ```
  ./wsadmin.sh -conntype SOAP
  ```

  and issue the above commands, then:

  ```
  set na [$AdminControl queryNames type=NodeSync,node=SY1,*]
  $AdminControl invoke $na sync
  ```

The process to prepare to run in 64-bit mode is similar for Network Deployment. First the deployment manager, node agent, and server needs to be configured. Save the configuration change and quit wsadmin. Then run the syncNode.sh script on the AppServer node to propagate the configuration change.

If the Deployment Manager and nodeagent are already running, you can connect using “wsadmin connect type SOAP” and issue the AdminTask commands as shown. Save the configuration changes. Then issue the set na and invoke $na sync as shown. Quit wsadmin.



IBM Software Group

Preparing to start a 64-bit server…

- Administration console: 64bit check box
  - Click server name, Configuration, 64bit Check box

* Short Name
BBOS001

Unique Id
BF340A910A22D680000000840000000209390F03

☑ Run in 64 bit JVM mode

  - Adjust the JVM min and max heap size
  - Click OK then Save
- ND mode considerations
  - After setting 64bit check box
    - Adjust the min and max JVM heap size
    - Click Apply and then Review
    - Click Synchronize Nodes
    - Then Save

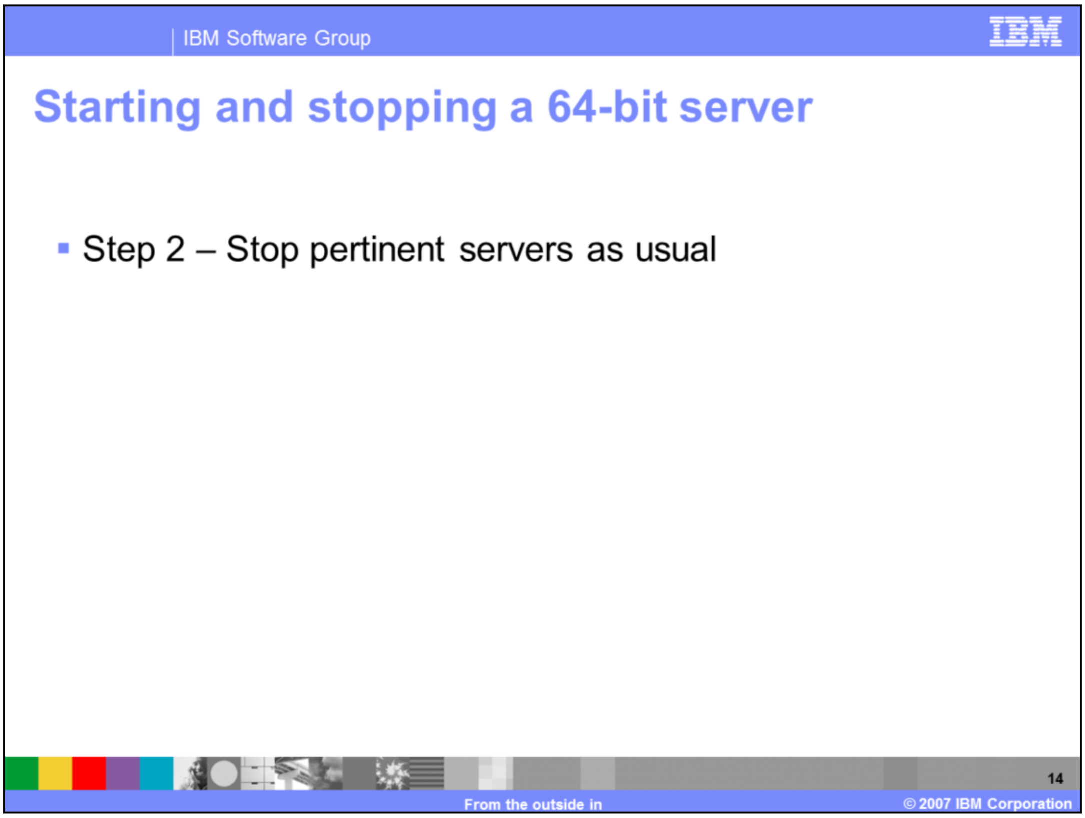
13

From the outside in

Another approach for configuring 64-bit mode, is a new configuration check box in the WebSphere administrative console. With your servers running and the console up, click System administration, the server you want to configure, then Configuration, and put a check in the “Run in 64bit mode” box.

Adjust the JVM minimum and maximum heap size to take advantage of the 64bit address space. Click OK then SAVE and the server is ready to run in 64-bit mode.

For Network Deployment mode, a couple of extra steps are required after setting the check box in order to synchronize each of the nodes. Remember to Save your changes.



IBM Software Group

Starting and stopping a 64-bit server

- Step 2 – Stop pertinent servers as usual

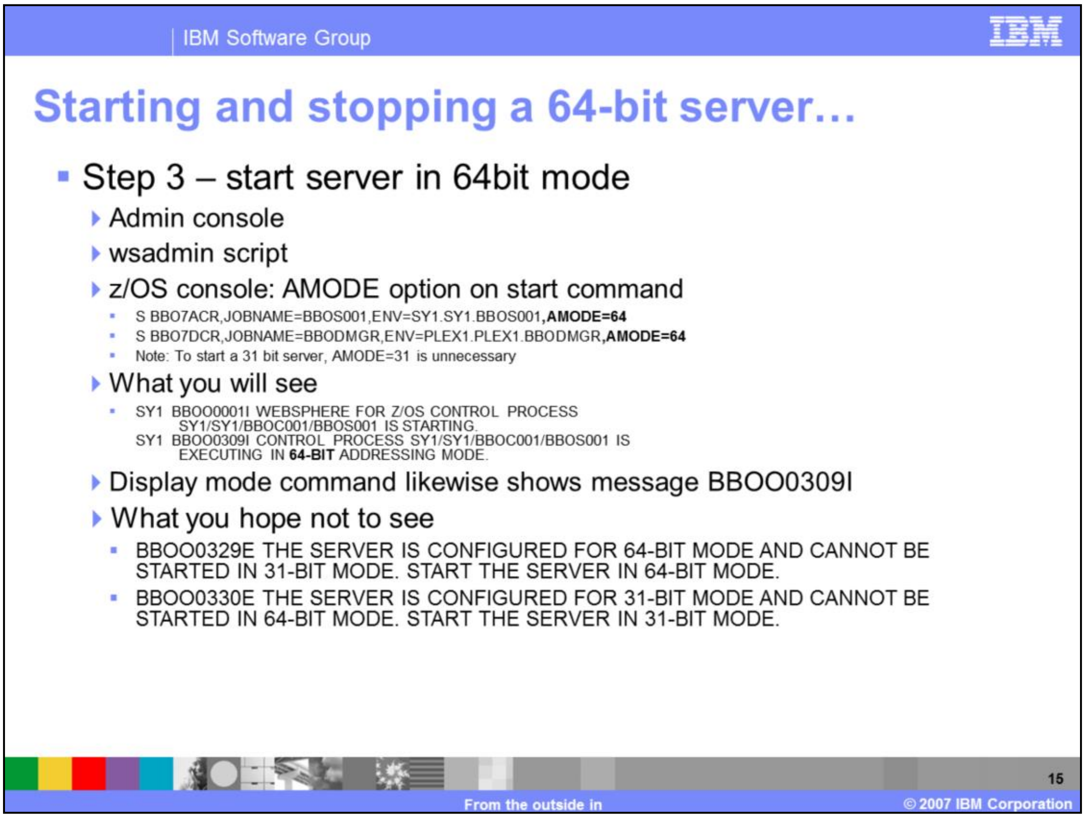
14

From the outside in

© 2007 IBM Corporation

After configuring the server, if it is already running, the server will need to be restarted in 64-bit mode.

Stopping servers is the same as in the past, using the command p <server name> on the z/OS operator console, clicking the administration console to stop it, or wsadmin scripting.



IBM Software Group

Starting and stopping a 64-bit server…

- Step 3 – start server in 64bit mode
  - Admin console
  - wsadmin script
  - z/OS console: AMODE option on start command
    - S BBO7ACR,JOBNAME=BBOS001,ENV=SY1.SY1.BBOS001,**AMODE=64**
    - S BBO7DCR,JOBNAME=BBODMGR,ENV=PLEX1.PLEX1.BBODMGR,**AMODE=64**
    - Note: To start a 31 bit server, AMODE=31 is unnecessary
  - What you will see
    - SY1 BBOO0001I WEBSPHERE FOR Z/OS CONTROL PROCESS SY1/SY1/BBOC001/BBOS001 IS STARTING.
SY1 BBOO0309I CONTROL PROCESS SY1/SY1/BBOC001/BBOS001 IS EXECUTING IN **64-BIT** ADDRESSING MODE.
  - Display mode command likewise shows message BBOO0309I
  - What you hope not to see
    - BBOO0329E THE SERVER IS CONFIGURED FOR 64-BIT MODE AND CANNOT BE STARTED IN 31-BIT MODE. START THE SERVER IN 64-BIT MODE.
    - BBOO0330E THE SERVER IS CONFIGURED FOR 31-BIT MODE AND CANNOT BE STARTED IN 64-BIT MODE. START THE SERVER IN 31-BIT MODE.

15

From the outside in

© 2007 IBM Corporation

Starting a 64-bit server can be accomplished in several ways.

First, using the administration console; there is no change with this process.

Second, using wsadmin or startServer scripts – again there is no change with this process.

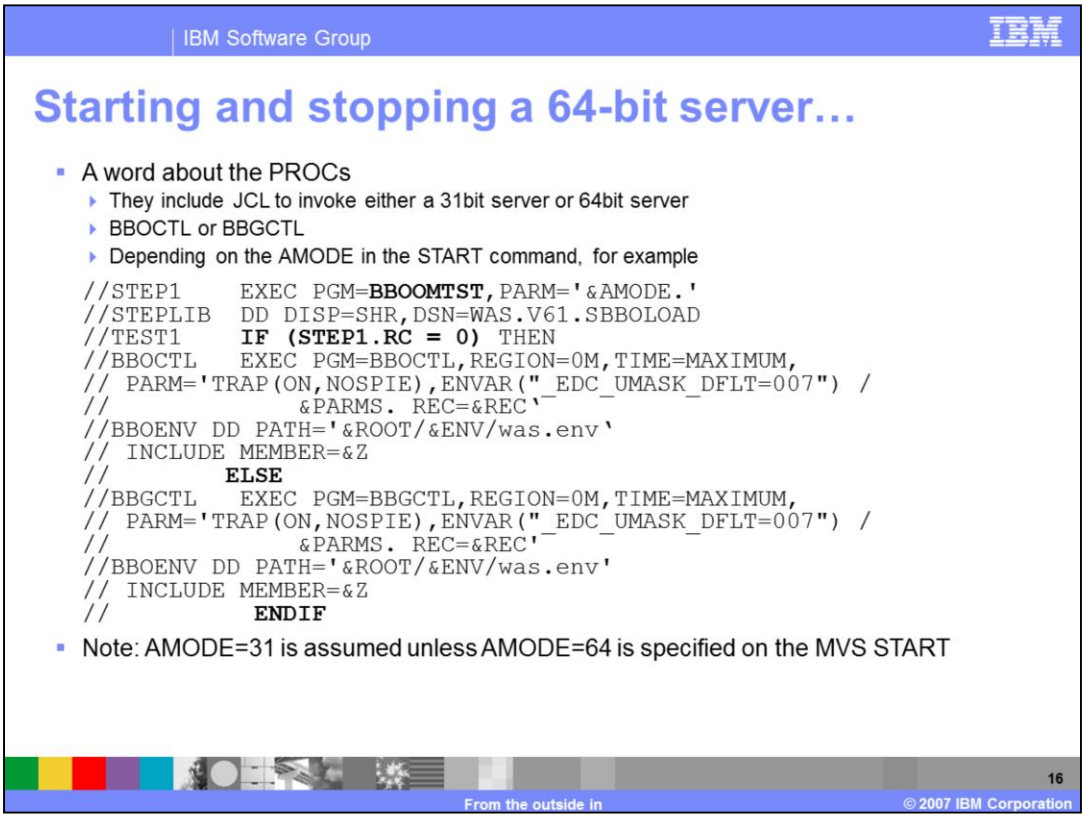
Finally, using the z/OS operation console, where a new parameter is required. Issue the start command as usual but now the AMODE option must be added to the end of the start command. The only useful operand is 64. Anything else – or leaving off AMODE altogether – will attempt to start the server in 31bit mode.

So as shown, the start command looks familiar except for the AMODE=64 added onto it.

What you hope to see is message BBOO0309I which tells you that the server you are starting is running in 64bit mode. This message is displayed for either mode that the server starts up in – whether 31-bit or 64-bit. Note that the daemon will continue to run in 31-bit mode so if the daemon needs to be started when you bring up a server, you will see this message for a 31bit daemon.

When you issue the display mode command, you will see message BBOO0309I for the server and it will show its running mode.

What you hope **not** to see is the mis-match message as shown here. It means the server was attempted to be started in a mode that it was not configured to run in. WebSphere Application Server V6.1 tries to catch this mismatch and halt the start gracefully.



Starting and stopping a 64-bit server…

- A word about the PROCs
  - They include JCL to invoke either a 31bit server or 64bit server
  - BBOCTL or BBGCTL
  - Depending on the AMODE in the START command, for example

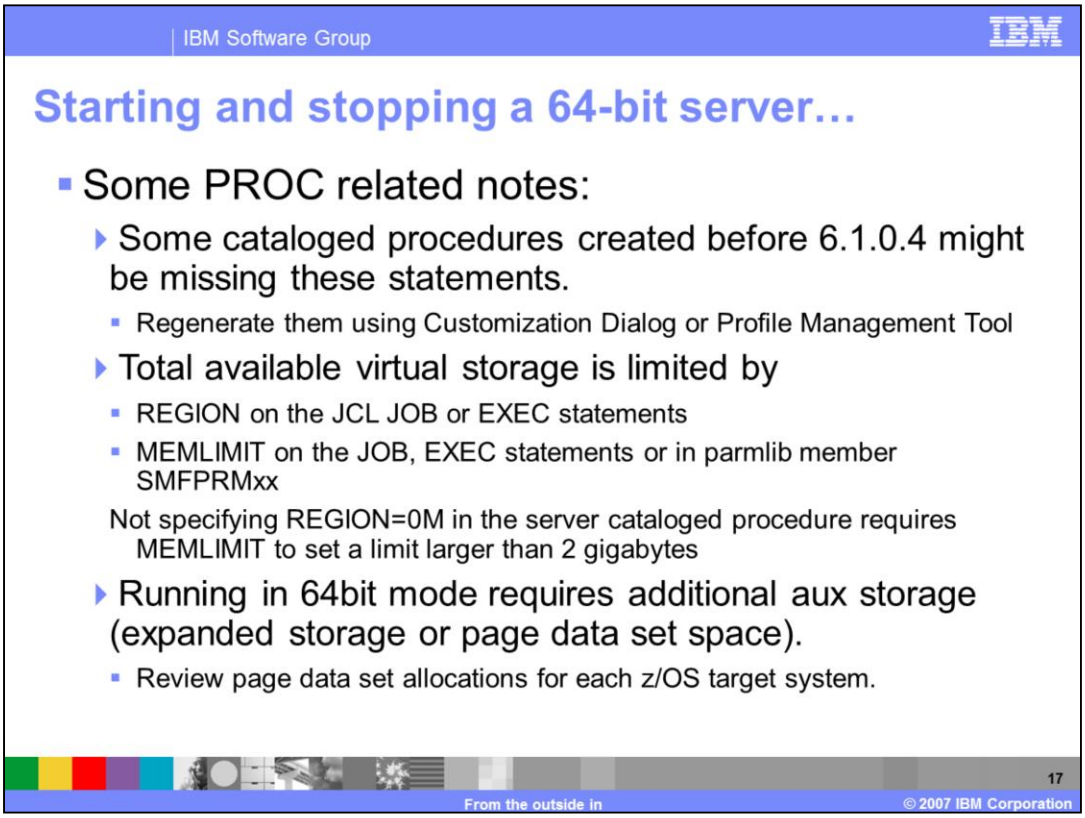
```
//STEP1     EXEC PGM=BBOOMTST,PARM='&AMODE.'
//STEPLIB   DD DISP=SHR,DSN=WAS.V61.SBBOLOAD
//TEST1     IF (STEP1.RC = 0) THEN
//BBOCTL    EXEC PGM=BBOCTL,REGION=0M,TIME=MAXIMUM,
//  PARM='TRAP(ON,NOSPIE),ENVAR("_EDC_UMASK_DFLT=007") /
//               &PARMS. REC=&REC'
//BBOENV DD PATH='&ROOT/&ENV/was.env'
//  INCLUDE MEMBER=&Z
//          ELSE
//BBGCTL    EXEC PGM=BBGCTL,REGION=0M,TIME=MAXIMUM,
//  PARM='TRAP(ON,NOSPIE),ENVAR("_EDC_UMASK_DFLT=007") /
//               &PARMS. REC=&REC'
//BBOENV DD PATH='&ROOT/&ENV/was.env'
//  INCLUDE MEMBER=&Z
//            ENDIF
```

- Note: AMODE=31 is assumed unless AMODE=64 is specified on the MVS START

Cataloged procedures for WebSphere Application Server for z/OS Version 6.1 include JCL statements that invoke either a 31bit server program (BBOCTL) or a 64bit server program (BBGCTL), depending on the AMODE value specified in the MVS START command.

BBOOMTST interrogates the AMODE parameter value and sets a return code appropriately: zero by default and non-zero if AMODE=64 was specified on the START command.

If the AMODE parameter is omitted or set to AMODE=31, the condition in the JCL runs BBOCTL. If AMODE=64 is specified on the MVS START command, then the 64-bit server program BB**G**CTL is run. The server program used must match the internal configuration setting for the server; if it does not, then the server terminates.



IBM Software Group

Starting and stopping a 64-bit server…

- Some PROC related notes:
  - Some cataloged procedures created before 6.1.0.4 might be missing these statements.
    - Regenerate them using Customization Dialog or Profile Management Tool
  - Total available virtual storage is limited by
    - REGION on the JCL JOB or EXEC statements
    - MEMLIMIT on the JOB, EXEC statements or in parmlib member SMFPRMxx

    Not specifying REGION=0M in the server cataloged procedure requires MEMLIMIT to set a limit larger than 2 gigabytes
  - Running in 64bit mode requires additional aux storage (expanded storage or page data set space).
    - Review page data set allocations for each z/OS target system.

17

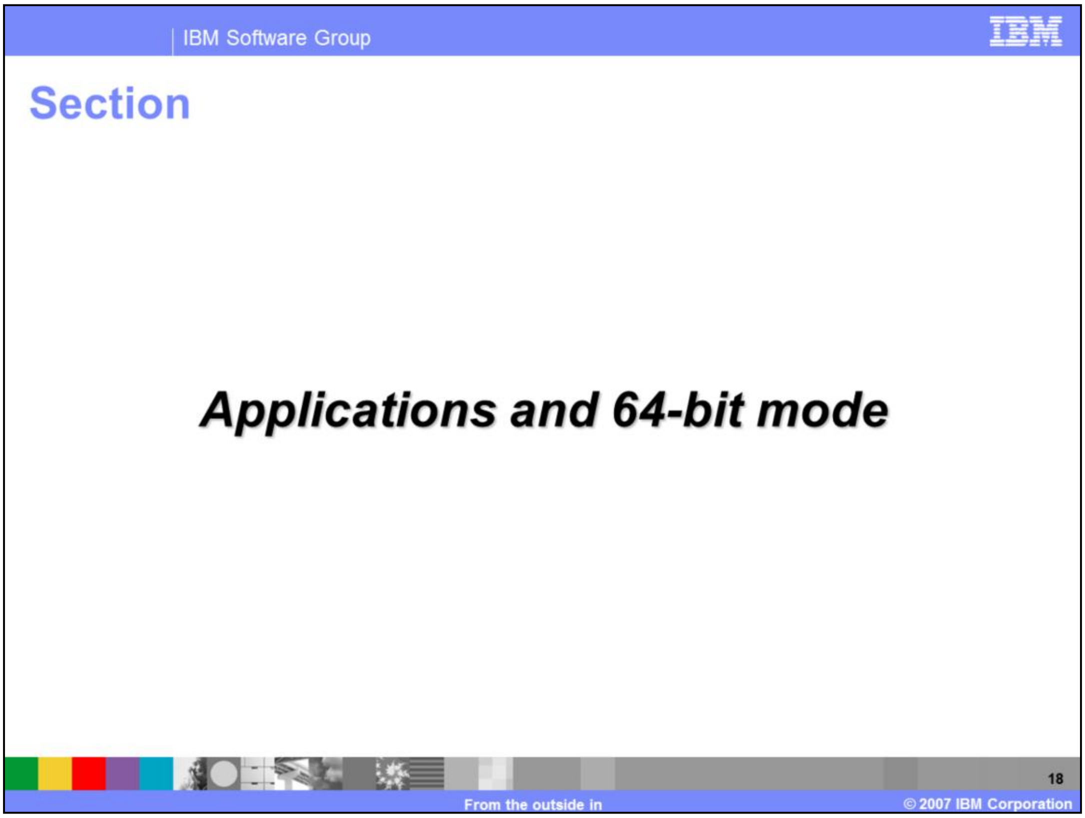
From the outside in

© 2007 IBM Corporation

In certain circumstances, WebSphere Application Server for z/OS Version 6.1 cataloged procedures that were created before Version 6.1.0.4 might be missing these statements. If so, regenerate the cataloged procedures for these servers using the Customization Dialog or Profile Management Tool and copy them to your system procedure library before converting the servers to 64-bit mode.

The amount of virtual storage that can be obtained by a particular server is determined by the REGION parameter, on the JCL JOB or EXEC statement, and by the MEMLIMIT parameter, on the JCL JOB or EXEC statement, or in the parmlib member SMFPRMxx. If you do not specify REGION=0M in the server cataloged procedure, you must use MEMLIMIT to set a virtual storage limit larger than the 2 gigabyte limit associated with 31-bit addressing mode. Also, make sure that your system exits do not limit the address space size for 64-bit servers inappropriately.

Running servers in 64-bit mode requires **additional auxiliary storage**, expanded storage or page data set space. Before running servers in 64-bit mode, review your page data set allocations for each z/OS target system. Add additional page data set space as needed, and monitor paging and page data set utilization.



IBM Software Group

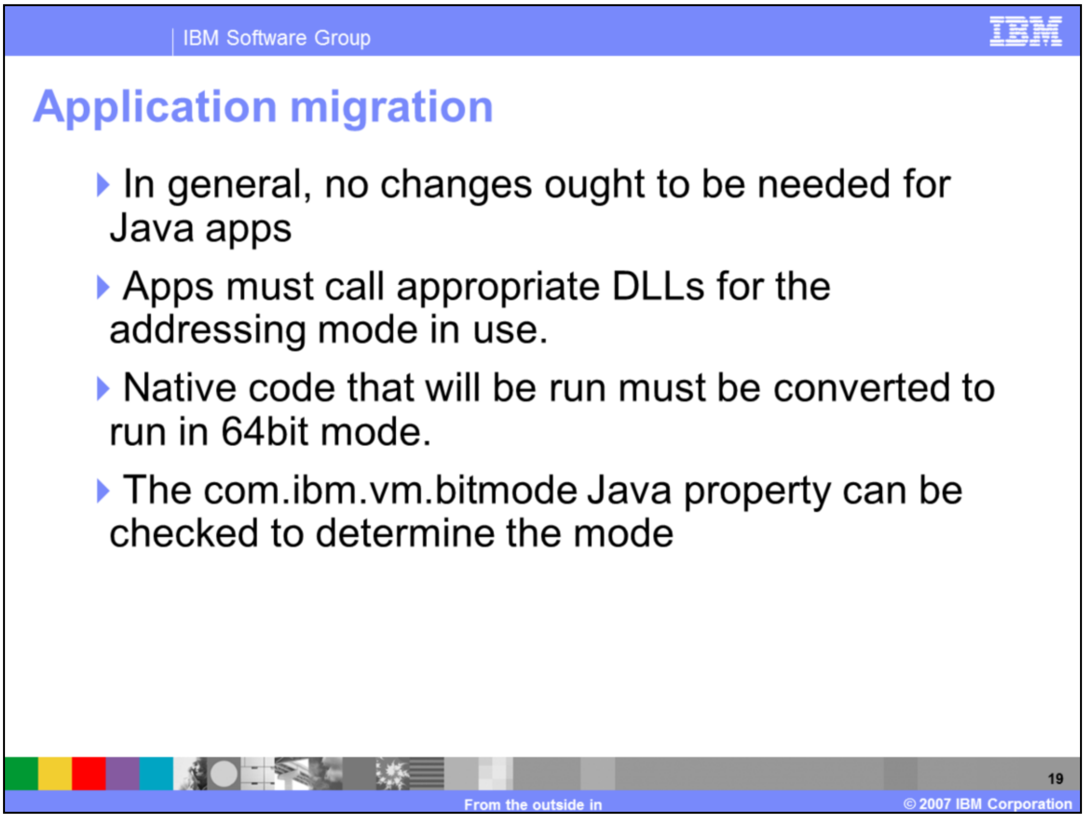
Section

Applications and 64-bit mode

18

From the outside in

This section will discuss applications and running in 64-bit mode on z/OS.



IBM Software Group

Application migration

- In general, no changes ought to be needed for Java apps
- Apps must call appropriate DLLs for the addressing mode in use.
- Native code that will be run must be converted to run in 64bit mode.
- The com.ibm.vm.bitmode Java property can be checked to determine the mode

19

From the outside in

© 2007 IBM Corporation

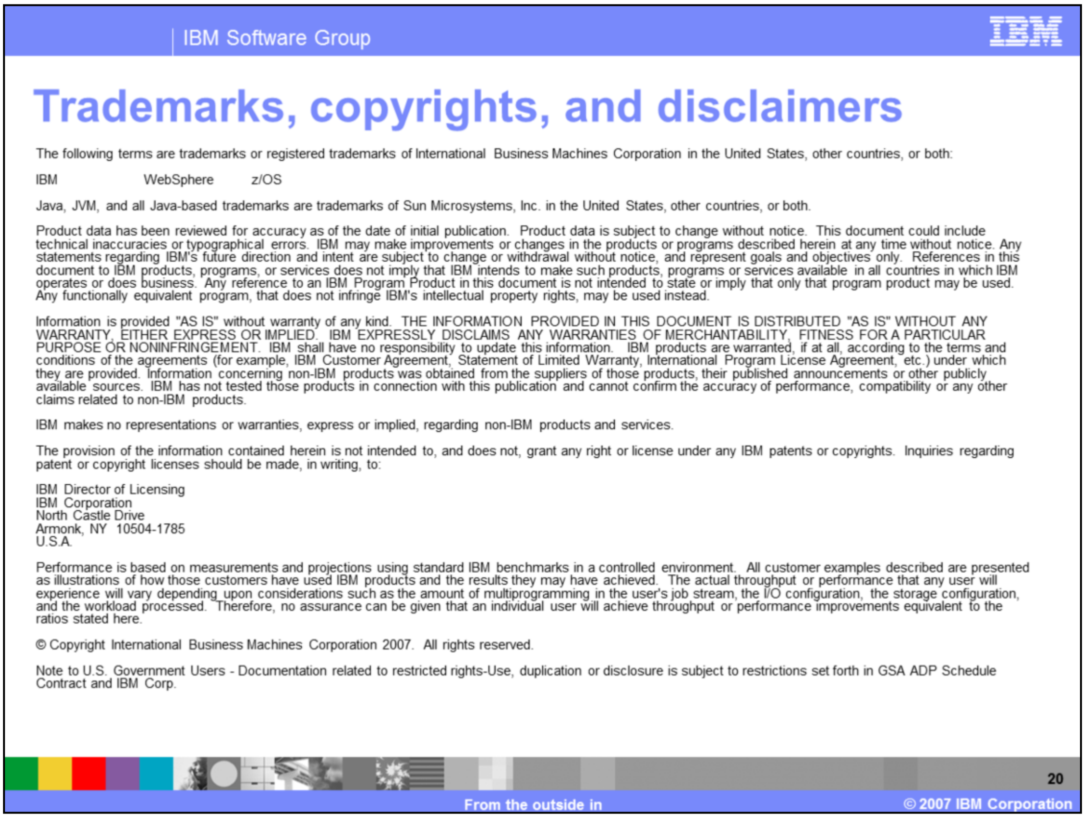
As mentioned before, the JVM is capable of running in 64-bit mode so there are no inherent changes required of your Java applications.

Applications must call appropriate DLLs and other native code for the addressing mode in use.

When converting an existing application server to 64-bit mode, any Java applications containing native code (for example C++ or Cobol) that will be run on the converted server must change to run in 64-bit mode.

The com.ibm.vm.bitmode Java property can be used by Java applications to determine the mode and load the correct 31-bit or 64-bit DLL to back native methods. An application containing a 31bit native module called under a 64bit runtime or an application containing a 64bit native module called in a 31bit runtime will probably abend.

Information on converting language-environment (LE) applications to run in 64-bit mode is available in the Language Environment Programming Guide for 64-bit Virtual Addressing Mode (SA22-7569).



IBM Software Group

Trademarks, copyrights, and disclaimers

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

IBM	WebSphere	z/OS

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

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

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

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

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

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

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

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.