



## IBM Tivoli Netcool/OMNibus V7.2.1 IEA

### *Creating a failover ObjectServer*

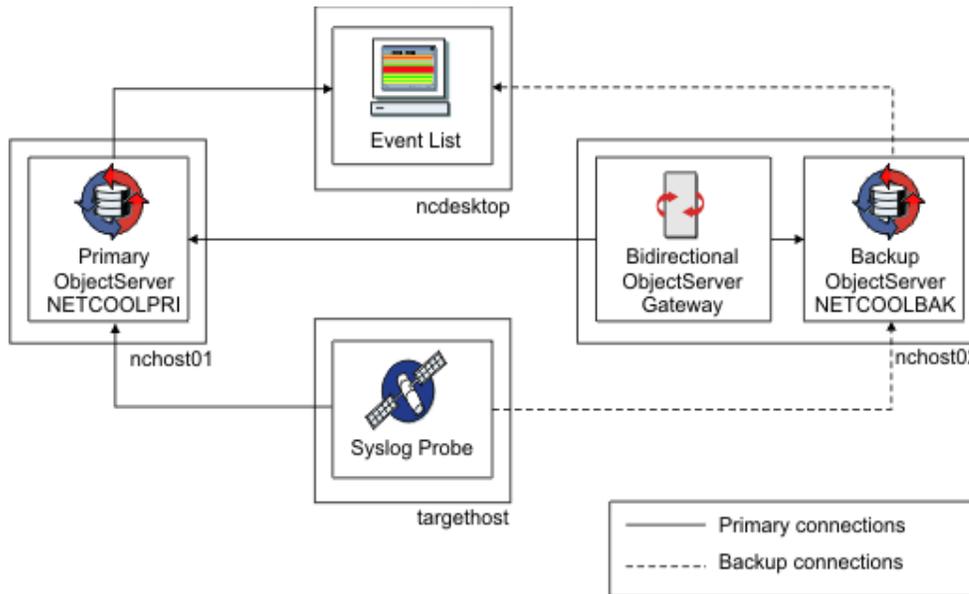
**Tivoli.** software



© 2009 IBM Corporation  
Updated August 5, 2009

Hello, and welcome to the OMNibus IBM Education Assistance module *Creating a failover ObjectServer*.

## Guidelines for creating a failover ObjectServer



Creating a failover ObjectServer

© 2009 IBM Corporation

The basic architecture for a failover pair consists of:

- A primary and backup ObjectServer
- A virtually defined ObjectServer (not shown in diagram)
- A bidirectional gateway and a
- Client entity configurations

## Guidelines for creating a failover ObjectServer

- The following actions are required to create the failover architecture:
  - ▶ Create the backup ObjectServer
  - ▶ Migrate the configuration from the primary to the backup ObjectServer
  - ▶ Define interface file entities
  - ▶ Set values within ObjectServer's property files
  - ▶ Create a bidirectional gateway between the primary and backup ObjectServers
  - ▶ Modify client entities properties to point to the virtual ObjectServer
  - ▶ Enable process control for gateway and ObjectServers

The actions required to create the failover architecture are as follows:

- Create the backup ObjectServer
- Migrate the configuration from the primary to the backup ObjectServer
- Define interface file entities
- Set values within ObjectServer's property files
- Create a bidirectional gateway between the primary and backup ObjectServers
- Modify client entities properties to point to the virtual ObjectServer
- Enable process control for gateway and ObjectServers

## The first two actions

- The first two actions in setting up a failover architecture are creating a backup ObjectServer and migrating the primary ObjectServer configuration to the backup ObjectServer. These actions are addressed in the following IBM Education Assistance modules:
  - ▶ IBM Tivoli Netcool/OMNIbus V7.2.1 Configuration  
*Creating and starting an ObjectServer*
  - ▶ IBM Tivoli Netcool/OMNIbus V7.2.1 Configuration  
*Using the nco\_confpack utility*
- Before continuing, you must create your BACKUP ObjectServer and migrate your PRIMARY ObjectServer configuration into the BACKUP ObjectServer.

The first two actions in setting up a failover architecture are creating a backup ObjectServer and migrating the primary's ObjectServer configuration to the backup ObjectServer. These actions are addressed in the following IBM Education Assistance modules:

- IBM Tivoli Netcool/OMNIbus V7.2.1 Configuration  
*Creating and starting an ObjectServer*
- IBM Tivoli Netcool/OMNIbus V7.2.1 Configuration  
*Using the nco\_confpack utility*

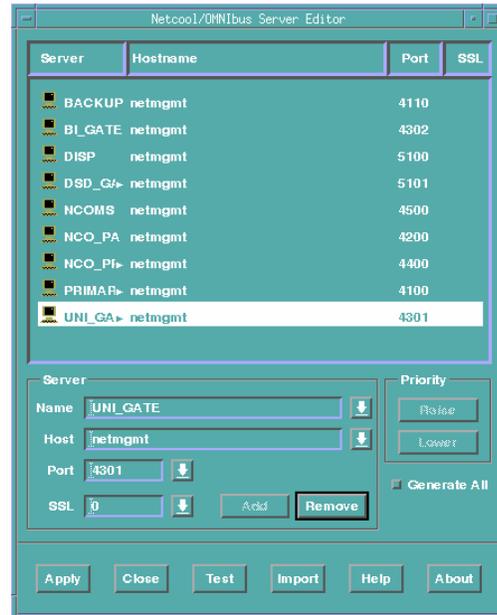
Before continuing, you must create your BACKUP ObjectServer and migrate your PRIMARY ObjectServer configuration into the BACKUP ObjectServer. Follow the processes outlined in the two IBM Education Assistance modules.

## Define interface file entities

- You must define your VIRTUAL ObjectServer and Failover gateway within the interface file.
- Your PRIMARY and BACKUP ObjectServers should already be defined.
- Launch the **nco\_xigen** GUI using the following commands:

```
cd $OMNIHOME/bin
./nco_xigen &
```

Your OMNIBus interface GUI will open.



You must define your VIRTUAL ObjectServer and Failover gateway within the interface file. Your PRIMARY and BACKUP ObjectServers should already be defined.

Launch the **nco\_xigen** GUI using the following commands:

```
Change directories to $OMNIHOME/bin
./nco_xigen &
```

Your OMNIBus interface GUI will open.

## Define interface file entities: virtual ObjectServer

To define your VIRTUAL ObjectServer entry within the OMNibus interface file, first enter a name for your VIRTUAL ObjectServer.

Server	Host	Port
PRIMAR	netmgmt	4100
UNI_GA	netmgmt	4301

<b>Server</b>		<b>Priority</b>	
Name	<input type="text" value="VIRTUAL"/>	<input type="button" value="None"/>	<input type="button" value="Lower"/>
Host	<input type="text" value="netmgmt"/>	<input type="checkbox"/> Generate All	
Port	<input type="text" value="4301"/>	<input type="button" value="Add"/>	<input type="button" value="Update"/>
SSL	<input type="text" value="0"/>		

To define your VIRTUAL ObjectServer entry within the OMNibus interface file, first enter a name for your VIRTUAL ObjectServer.

## Define interface file entities: virtual ObjectServer

Next, enter the port number and host information. Use the same values as your PRIMARY ObjectServer's port and host. Click **ADD** and then click **APPLY**.

Name	Host	Port
PRIMARY	netmgmt	4100
UNL_GATE	netmgmt	4301

Server

Name:

Host:

Port:

SSL:

Priority:

Generate All

Next, enter the port number and host information. Use the same values as your PRIMARY ObjectServer's port and host. Click **ADD** and then click **APPLY**.

## Define interface file entities: virtual ObjectServer

After you have added the VIRTUAL ObjectServer, click the newly created VIRTUAL ObjectServer name.

UNI_GATE	netmgmt	4301
<b>VIRTUAL</b>	netmgmt	4100
BACKUP	nethost	4110

Server

Name:  ↓

Host:  ↓

Port:  ↓

SSL:  ↓

Priority

Generate All

Creating a failover ObjectServer © 2009 IBM Corporation

After you have added the VIRTUAL ObjectServer, click the name of the newly created VIRTUAL ObjectServer.

## Define interface file entities: virtual ObjectServer

- In the port identifier, change the port number from the previously entered primary port number to the same port number as your BACKUP ObjectServer's listening port.
- Additionally, change the host identifier from the primary's host to the BACKUP ObjectServer's host name.

UNI_GATE	netmgmt	4301
VIRTUAL	netmgmt	4100
BACKUP	nethost	4110

Server

Name: VIRTUAL

Host: nethost

Port: 4110

SSL: 0

Priority: None, Lower

Generate All

Add Update

Apply Close Test Import Help About

In the port identifier, change the port number from the previously entered primary port number to the same port number as your BACKUP ObjectServer's listening port. Additionally, change the host identifier from the primary's host to the BACKUP ObjectServer's host name.

## Define interface file entities: virtual ObjectServer

- Click **ADD** and **APPLY**.
- You will see your newly created **BACKUP** ObjectServer instance below your **VIRTUAL** ObjectServer identifier.

Instance Name	Host	Port
NCO_PROXY	netmgmt	4400
PRIMARY	netmgmt	4100
UNI_GATE	netmgmt	4301
VIRTUAL	netmgmt	4100
Backup1:	nethost	4110

Server	Priority
Name: VIRTUAL	Raise
Host: nethost	Lower
Port: 4110	<input type="checkbox"/> Generate All
SSL: 0	<b>Add</b> Remove

<b>Apply</b>	Close	Test	Import	Help	About
--------------	-------	------	--------	------	-------

Click **ADD** and **APPLY**.

Observe your newly created **BACKUP** ObjectServer instance below your **VIRTUAL** ObjectServer identifier.

## Define interface file entities: bidirectional gateway

- Define a bidirectional gateway within your **nco\_xigen** interface. As a general guideline, create your bidirectional gateway within or near your BACKUP ObjectServer host.
- The gateway requires a unique port number within the selected host. A gateway's default port value is 4300.
- The naming convention you use in this GUI must be carried over to the naming convention you will use to create the gateway's property file.

For example, if you name the bidirectional gateway **FAIL\_GATE**, you must use that name in the creation of your related bidirectional gateway file, discussed later.

- Define a bidirectional gateway within your **nco\_xigen** interface. As a general guideline, create your bidirectional gateway within or near your BACKUP ObjectServer host.
- The gateway requires a unique port number within the selected host. A gateway's default port value is 4300.
- The naming convention you use in this GUI must be carried over to the naming convention you will use to create the gateway's property file.

For example, if you name the bidirectional gateway **FAIL\_GATE**, you must use that name in the creation of your related bidirectional gateway file, discussed later.

## ObjectServer property values

- The only required ObjectServer property value change is within the BACKUP ObjectServer's property file. Navigate to the **\$OMNIHOME/etc** directory and open the **BACKUP.props** file for editing. After you open the file, edit the value of **BackupObjectServer** from **FALSE** to **TRUE**. Use the following commands:

```
cd $OMNIHOME/etc
```

```
vi BACKUP.props
```

Then change FALSE to TRUE:

```
BackupObjectServer: FALSE
```

```
BackupObjectserver: TRUE
```

The only required ObjectServer property value change is within the BACKUP ObjectServer's property file. Navigate to the **\$OMNIHOME/etc** directory and open the **BACKUP.props** file for editing. After you open the file, edit the value of **BackupObjectServer** from **FALSE** to **TRUE**.

Use the following commands:

```
change directories to $OMNIHOME/etc
```

```
and use vi to edit the BACKUP.props file.
```

## Bidirectional gateway creation

A key component in the failover and failback architecture is the bidirectional gateway. To create the gateway configuration, follow these steps:

1. On the BACKUP ObjectServer's computer (**nethost**), create the directory **\$NCHOME/omnibus/gates/FAIL\_GATE**.
2. Copy all of the files in **\$NCHOME/omnibus/gates/objserv\_bi** to the **\$NCHOME/omnibus/gates/FAIL\_GATE** directory.
3. Rename the **\$NCHOME/omnibus/gates/FAIL\_GATE/objserv\_bi.map** file to **FAIL\_GATE.map**.
4. Edit the **FAIL\_GATE.map** file. The fields and field order in the mapping must match the **alerts.status** table exactly in the PRIMARY and BACKUP ObjectServers.

A key component in the failover and failback architecture is the bidirectional gateway. To create the gateway configuration, follow these steps:

1. On the backup computer, create the directory **\$NCHOME/omnibus/gates/FAIL\_GATE**.
2. Copy all of the files in **\$NCHOME/omnibus/gates/objserv\_bi** to the **\$NCHOME/omnibus/gates/FAIL\_GATE** directory.
3. Rename the **objserv\_bi.map** file to **FAIL\_GATE.map**.
4. Edit the **FAIL\_GATE.map** file. The fields and field order in the mapping must match the **alerts.status** table exactly in the primary and backup ObjectServers.

## Bidirectional gateway creation

Observe the following example of field mappings between the PRIMARY and BACKUP ObjectServers.

```
CREATE MAPPING StatusMap
(
  'Identifier'      = '@Identifier'  ON INSERT ONLY,
  'Node'           = '@Node'        ON INSERT ONLY,
  'NodeAlias'     = '@NodeAlias'   ON INSERT ONLY,
  ...
  ...
  'CustomerID'    = '@CustomerID'   ON INSERT ONLY,
  'CustomerContact' = '@CustomerContact' ON INSERT ONLY,
  'ReferenceCode' = '@ReferenceCode' ON INSERT ONLY,
  'ServerName'   = '@ServerName'   ON INSERT ONLY,
  'ServerSerial' = '@ServerSerial'  ON INSERT ONLY
```

Observe the following example of standard field and custom field mappings between the PRIMARY and BACKUP ObjectServers. Custom fields are in bold font.

## Bidirectional gateway creation

- Rename the **\$NCHOME/omnibus/gates/FAIL\_GATE/objserv\_bi.props** file to **FAIL\_GATE.props**.
- Edit the entries in the **FAIL\_GATE.props** file as seen in the following slide.

Rename the **\$NCHOME/omnibus/gates/FAIL\_GATE/objserv\_bi.props** file to **FAIL\_GATE.props**.

Edit the entries in the **FAIL\_GATE.props** file as seen in the following slide.

## Properties

**Common gateway properties:**

```
Gate.MapFile      : '$OMNIHOME/etc/BI_GATE.map'  
Gate.StartupCmdFile : '$OMNIHOME/etc/BI_GATE.startup.cmd'
```

**Bidirectional ObjectServer gateway properties:**

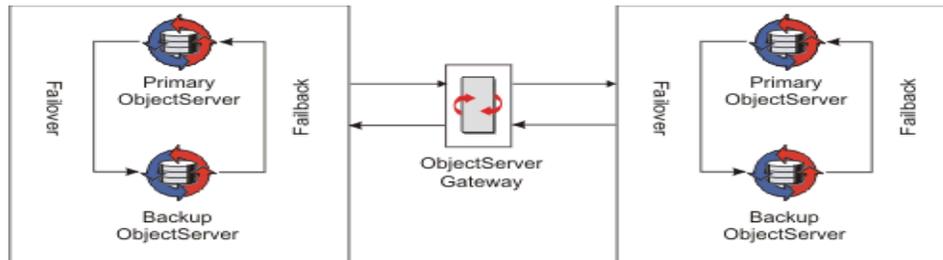
```
Gate.ObjectServerA.Server : 'PRIMARY'  
Gate.ObjectServerA.StatusTableName : 'alerts.status'  
Gate.ObjectServerA.JournalTableName : 'alerts.journal'  
Gate.ObjectServerA.DetailsTableName : 'alerts.details'  
Gate.ObjectServerA.TblReplicateDefFile : '$OMNIHOME/etc/BI_GATE.objectservera.tblrep.def'  
Gate.ObjectServerA.Description : 'BI_GATEA'  
Gate.ObjectServerB.Server : 'SECONDARY'  
Gate.ObjectServerB.StatusTableName : 'alerts.status'  
Gate.ObjectServerB.DetailsTableName : 'alerts.details'  
Gate.ObjectServerB.JournalTableName : 'alerts.journal'  
Gate.ObjectServerB.TblReplicateDefFile : '$OMNIHOME/etc/BI_GATE.objectserverb.tblrep.def'  
Gate.ObjectServerB.Description : 'BI_GATEB'  
Gate.Resync.Enable : TRUE  
Gate.Resync.Type : 'NORMAL'  
Gate.Resync.Master : ''
```

Use these values as a general guideline. You must define the **Gate.ObjectServerA.Server** with your PRIMARY ObjectServer definition used in the interface file. You must also define the **Gate.ObjectServerB.Server** as your previously created BACKUP ObjectServer.

## Bidirectional gateway creation: enable failback

To enable failback, in the gateway properties file you must set the **Gate.ObjectServerA.Failback** property to **TRUE** (if ObjectServer A has a backup ObjectServer) and **Gate.ObjectServerB.Failback** property to **TRUE** (if ObjectServer B has a backup ObjectServer).

When the primary ObjectServer has been detected again, the gateway automatically fails back to the primary ObjectServer. To specify the frequency with which the gateway polls the failed ObjectServer, set **Gate.ObjectServerA.FailbackTimeout** and **Gate.ObjectServerB.FailbackTimeout**.



Creating a failover ObjectServer

© 2009 IBM Corporation

To enable failback, in the gateway properties file, you must set the **Gate.ObjectServerA.Failback** property to **TRUE** (if ObjectServer A has a backup ObjectServer) and **Gate.ObjectServerB.Failback** property to **TRUE** (if ObjectServer B has a backup ObjectServer).

When the primary ObjectServer has been detected again, the gateway automatically fails back to the primary ObjectServer.

To specify the frequency with which the gateway polls the failed ObjectServer, set **Gate.ObjectServerA.FailbackTimeout** and **Gate.ObjectServerB.FailbackTimeout**.

## Bidirectional gateway creation

- The FAIL\_GATE gateway can have a number of other properties set. The property values presented reflect a minimum set of values to make the gateway functional.
- Refer to the *IBM Netcool/OMNIBus Probe and Gateway Guide* (SC23-6387) for further details and to enable security parameters.

The FAIL\_GATE gateway can have a number of other properties set. The property values presented reflect a minimum set of values to make the gateway functional.

Refer to the *IBM Netcool/OMNIBus Probe and Gateway Guide* (SC23-6387) for further details and to enable security parameters.

## Client property values

A ***client*** is any entity that connects to an ObjectServer. The entity value changes presented next will reflect values within unidirectional gateways, probes, and desktops.

A ***client*** is any entity that connects to an ObjectServer. The entity value changes presented next will reflect values within unidirectional gateways, probes, and desktops.

## Configuring client processes for failover

- All client processes can now connect to the virtual pair.
  - ▶ For desktops, select the **VIRTUAL** server from the ObjectServer list.
- Gateways will have the Reader or Writer connection properties set as **VIRTUAL** where necessary. The most common values are:

```
Gate.Reader.Server : 'VIRTUAL'
Gate.Reader.Username : 'root'
Gate.Reader.Password : ''
```

- Probes must be configured in the **<probename>.props** file:

```
Server : 'PRIMARY'
ServerBackup : 'BACKUP'
NetworkTimeout : integer value less than PollServer value
PollServer : integer value
AutoSAF : 1
```

**Note:** The FAIL\_GATE bidirectional gateway must reference the real ObjectServers.

All client processes can now connect to the virtual pair.

For desktops, select the **VIRTUAL** server from the ObjectServer list.

Gateways will have the Reader or Writer connection properties set as **VIRTUAL** where necessary. The most common values are:

```
The Reader.Server
The Reader.Username and
The Reader.Password
```

Probes must be configured in the **<probename>.props** file. You must set the Server, ServerBackup, NetworkTimeout, PollServer, and Auto store and forward values.

Note that the FAIL\_GATE bidirectional gateway must reference the real ObjectServers.

## Client property values

- Probes rules files might differ due to the specific nature of the different probes. However, if you point the probe away from the PRIMARY ObjectServer and to the VIRTUAL ObjectServer identifier, failover will also resolve the probes focus during a failover event.
- **Note:** The interface file on each separate host, where probes may reside, must have the definitions for the PRIMARY, BACKUP, and VIRTUAL ObjectServers.

Probes rules files might differ due to the specific nature of the different probes. However, if you point the probe away from the PRIMARY ObjectServer and to the VIRTUAL ObjectServer identifier, failover will also resolve the probes focus during a failover event.

**Note:** The interface file on each separate host, where probes may reside, must have the definitions for the PRIMARY, BACKUP, and VIRTUAL ObjectServers.

## Define failover in process control

To define the BACKUP ObjectServer and failover gateway in process control, you can use the steps outlined in the IBM Education Assistance module OMNibus 7.2.1 Automation, process automation to launch external actions.

To define the BACKUP ObjectServer and failover gateway in process control, you can use the steps outlined in the IBM Education Assistance module OMNibus 7.2.1 Automation, process automation to launch external actions.

## Define failover in process control

You must include the following entry in the **nco\_pa.conf** file to enable the bidirectional gateway.

```
nco_process 'Bi_Gate'
{
  Command '$NCHOME/omnibus/bin/nco_g_objserv_bi -name FAIL_GATE' run as 0
  Host='nchost02'
  Managed=true
  RestartMsg='The bidirectional gateway has been restarted'
  AlertMsg='The bidirectional gateway has gone down'
  RetryCount=5
  ProcessType=PaPA_AWARE
}
nco_service 'Bak_OS'
{
  ServiceType=Master
  ServiceStart=Auto
  process 'Bak_ObjectServer' NONE
  process 'Bi_Gate' 'Bak_ObjectServer'
}
```

You must include the following entry in the **nco\_pa.conf** file to enable the bidirectional gateway. Your definition in the **nco\_service** can differ, depending upon your particular implementation. You might define the gateway processes to start during automation initiation.

## The final steps

- To complete the failover architecture, you must:
  - ▶ Ensure that both the PRIMARY and BACKUP ObjectServers are operational
  - ▶ Ensure that the FAIL\_GATE gateway is operational
  - ▶ Ensure that process control is operational in your architecture
- In the Netcool Foundation Administration GUI, navigate to the running process control GUI and verify that all elements are active.
- To test functionality, in your development environment, bring down the PRIMARY ObjectServer and ensure that your desktop, probes, and unidirectional gateways switch over to the BACKUP ObjectServer.

To complete the failover architecture, you must:

- Ensure that both the PRIMARY and BACKUP ObjectServers are operational
- Ensure that the FAIL\_GATE gateway is operational
- Ensure that process control is operational in your architecture

In the Netcool Foundation Administration GUI, navigate to the running process control GUI and verify that all elements are active. This process is also described in the IBM Education Assistance module under Automation.

To test functionality, in your development environment, bring down the PRIMARY ObjectServer and ensure that your desktop, probes, and unidirectional gateways switch over to the BACKUP ObjectServer.

## Training roadmap for *Tivoli Netcool/OMNIbus*

[http://www.ibm.com/software/tivoli/education/edu\\_prd.html](http://www.ibm.com/software/tivoli/education/edu_prd.html)



For further training, refer to the following link:

[http://www.ibm.com/software/tivoli/education/edu\\_prd.html](http://www.ibm.com/software/tivoli/education/edu_prd.html)

## Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, and the following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

Netcool            Tivoli

If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of other IBM trademarks is available on the Web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>

Other company, product, or 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 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.

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 NON-INFRINGEMENT. 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.

© Copyright International Business Machines Corporation 2009. 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.

This concludes this module.