



Omnibus Failover Installation Cookbook

GSMA – Netcool-based Event Management

Author: Florence Rouanet-Rose

Customer: Global Cross Services Technology Integration

M&E project

17/10/2019

Document version: 10.1.1.0

Document status: FINAL

Document History

Document Location

This is a snapshot of an on-line document. Paper copies are valid only on the day they are printed. Refer to the author if you are in any doubt about the currency of this document.

Revision History

Date of this revision: 2019/09/17	Date of next revision (date)
-----------------------------------	--

Revision Number	Revision Date	Summary of Changes	Changes marked
1.0.0	07/09/2010	Document Creation	(N)
1.1.0	10/11/2010	Final version	
1.1.1	15/11/2010	Correct setting of obsrvr variable in Primary Object Server installation. Add information on possible error messages received when running installation SQL scripts.	
1.1.2	02/12/2010	Corrected user for gateway setting chapter rs. Added new ObjectServer chapter. Added new Secure SSL Communication setting chapter.	
1.1.3	27/01/2011	Update the Proxy Server Settings chapters : .	
1.1.5	21/03/2011	Add Upgrade procedure in failover mode	
1.1.6	09/05/2011	Correct the AGG_GATE.props file setting in 3.5 chapter (defect 166809)	
3.4	03/11/2011	Include failover setting for the ODBC gateway	
3.5.1	30/04/2012	untar the Failover package in /opt/IBM/GSMA instead of /opt/IBM/GSMA/failover	
3.6	15/10/2012	Added the DelayToTransferFile new parameter for ODBC gateway failover solution.	
3.6.1	5.04.2013	Added the SSL & FIPS configuration.	
3.6.1a	06/13/13	Corrected the section 1.1 talks about the nco_sql command being used by automation scripts	
3.7	07/16/13 07/19/13	Changed "fail back" option to "failover" for Object-Server datasource definition. Correct installation steps for authorisation on \$NCHOME/omnibus/var/nco_g_odbc directory and ssh settings correct Path in command in chapter 6.2.1 and 9.1.2	
3.7a	30/09/2013	Changed 3.7 Reconfigure Aggregation gateway for agggateway user. Changed Resync type to Normal and added Gate.Resync.Prefered parameter. Changes in AGG_GATE.tblrep.def file	
9.0	2014/04/10 2014/04/23 2014/08/10	Change for SSL key length change to 2048 JDBC gateway as a replacement for ODBC gateway Revise scripts for different passwords between Operating system and Objectserver	



Enablement and Exploitation of GSMA

9.1.0.1	2015/03/23	Correct the table of content Correct the setting of the Shutdown script. Correct the installation steps for the silent install file.	
9.3.0.0	2016/06/08	Corrected references to external cookbook Added procedure for aggateway user creation	
9.5.1.0	2017/04/28	gsmaTDWGatewayMonitoring.conf.sample file updated to have parameter NCHOME	
9.5.7.0	2017/10/10	Update for netcool failover on RHEL 7.3	
9.8.3.0	2018/12/27	Updated IBM's new password policy rule as note	
9.8.3.0	2019/01/02	Updated Gate.Resync.Type to MINIMAL when updating AGG_GATE.props with aggateway user, at chapter 3.7.	
9.9.1	2019/02/20	Updated Manual FIPS Configuration steps	
10.0.2	2019/08/30	Modified target includes statement - Defect 447492	
10.1.0	2019/09/17	Fix typos and remove old GSMA webpage references - Defect 447492	
10.1.1.0	17/10/2019	Updated all the sections, chapter numbers that are mismatching and restored the navigation link to chapters. Also, updated the secure communication steps from webgui 7.x to 8.1 and impact 5.x to 7.x in section 6.2 and 6.3 as per defect 559107	

Contributors

Contributor Name	Title
Eric Champy	GSMA Event Management Architect
Mathieu Carpentier	SMI IT Specialist

Approvals

Distribution

This document is available on the latest Baseline under [Packages wiki page](#)

Table of contents

1.1 Failover configuration.....	8
1.2 Configuring server communication information.....	10
1.2.1 Processing.....	10
1.2.2 Naming Convention.....	13
1.2.3 Communication port setting.....	13
2.1 Installing and configuring the Primary ObjectServer.....	14
2.2 Installing and configuring the Backup ObjectServer.....	14
2.2.1 Install the Backup ObjectServer.....	14
2.2.2 Configure the Backup ObjectServer.....	14
3.1 Configuring Failover on the primary ObjectServer.....	18
3.2 Configuring controlled shutdown of the Primary ObjectServer.....	19
3.3 Configure Failover on the Backup Object Server.....	20
3.4 Defining User ID and roles for Aggregation Gateway on Primary and Backup Object Servers.....	21
3.5 Configuring the Aggregation Gateway on Backup Object Server.....	23
3.6 Configure Process Agent for Aggregation Gateway on Backup Object Server.....	24
3.7 Reconfigure Aggregation gateway for agggateway user.....	26
4.1 Primary Object Server configuration for SSL.....	27
4.1.1 Create Certificate Key Database.....	27
4.1.2 Create a CA certificate (on the ObjectServer).....	28
4.1.3 Setting up SSL on Primary Object Server.....	29
4.2 Backup Object Server configuration for SSL.....	30
4.2.1 Setting up SSL Certificate Database.....	30
4.2.2 CA certificate.....	31
4.2.3 Setting-up SSL on the ObjectServer.....	31
5.1 Primary Object Server configuration for SSL & FIPS.....	33
5.1.1 FIPS configuration.....	33
5.1.2 Create Certificate Key Database.....	34
5.1.3 Create a CA certificate (on the ObjectServer).....	34
5.1.4 Setting up SSL on Primary Object Server.....	35
5.2 Backup Object Server configuration for SSL & FIPS.....	37

5.2.1	FIPS configuration.....	37
5.2.2	Setting up SSL Certificate Database.....	38
5.2.3	CA certificate.....	38
5.2.4	Setting-up SSL on the ObjectServer.....	39
6.1	Configure Probe to route events to the virtual aggregation pair.....	41
6.1.1	Setting up SSL Certificate Database.....	42
6.1.2	Setting up SSL & FIPS Certificate Database.....	42
6.2	Configure WebGUI to connect on the virtual aggregation pair.....	44
6.2.1	Secure Communication between WebGUI and aggregation pair.....	45
6.3	Configure Impact to connect to the virtual aggregation pair.....	49
6.3.1	Using SSL Connection to the ObjectServer.....	51
6.4	Other Netcool Components.....	53
6.4.1	Unidirectional ObjectServer Gateways:.....	54
6.4.2	Bidirectional ObjectServer Gateways.....	54
6.4.3	Event lists.....	54
6.5	Configuring proxy servers for failover.....	54
6.5.1	Principle.....	54
6.5.2	Proxy server configuration.....	56
7.1	Overview.....	59
7.2	Configuring backup EIF probe.....	61
7.2.1	Setting up SSL & FIPS Certificate Database.....	61
7.3	Configuring new Event destination on TEMS.....	62
8.1	Overview of the GSMA solution.....	64
8.1.1	Detailed TDW Gateway Failover solution content.....	66
8.1.2	SSH tunnel.....	66
8.1.3	Solution limitations.....	67
8.2	TDW Gateways settings.....	67
8.3	Settings on system hosting TDW Primary Gateway.....	68
8.3.1	Users Specifications.....	68
8.4	Settings on system hosting TDW Failover Gateway.....	69
8.4.1	Users Specifications.....	69
8.4.2	SSH Tunnel settings to transfer files.....	69
8.4.3	Install and Configure the GSMA TDW Gateway Monitoring script....	72
8.5	Backup ObjectServer settings.....	73
9.1	Upgrade ObjectServer Aggregation pair.....	74



9.1.1 Pre/Co Requisites.....	75
9.1.2 Upgrade GSMA EDM and Related Automation on ObjectServer.....	76
9.2 Upgrade GSMA Failover code for Aggregation gateway.....	76
9.3 Upgrade GSMA EIF Probes code in failover mode.....	76
9.3.1 Error messages in Gateway log.....	77
9.3.2 Error messages in Backup Object Server log.....	77
Appendix A. List of default OMNibus triggers used during failover.....	77

List of figures

Figure 1 - Overall OMNibus failover configuration.....	11
Figure 2 - ObjectServers communications during failover – Set-up.....	12
Figure 3 - ObjectServers communications during failover – Primary Up.....	13
Figure 4 - Impact setup on an OMNibus failover pair.....	50
Figure 5 - Failover at probe/proxy level.....	54
Figure 6 – EIF Probe failover - Flow.....	59



Introduction

This document applies to Netcool Event Management domain within GSMA Project.

For clarity reasons, some extracts of Tivoli Netcool Omnibus documentation about failover configuration have been included in this documentation. They will appear in *italic*.

1.ObjectServer Failover overview

1.1 Failover configuration

The failover configuration is a requirement for high availability, and is based around the aggregation layer of the standard multitiered architecture. In its simplest configuration, the failover configuration consists of a primary and a backup ObjectServer that are connected by a bidirectional ObjectServer Gateway in the aggregation layer, with no collection or display layers connected.

Failover configuration is different from the load-balancing configuration. In a failover configuration only one of the two ObjectServers involved in the solution is acting as primary at a given time. When primary Object Server is running, the Backup ObjectServer is running in stand by, waiting for the primary to shut down.

The below illustrates an OMNibus ObjectServer failover configuration:

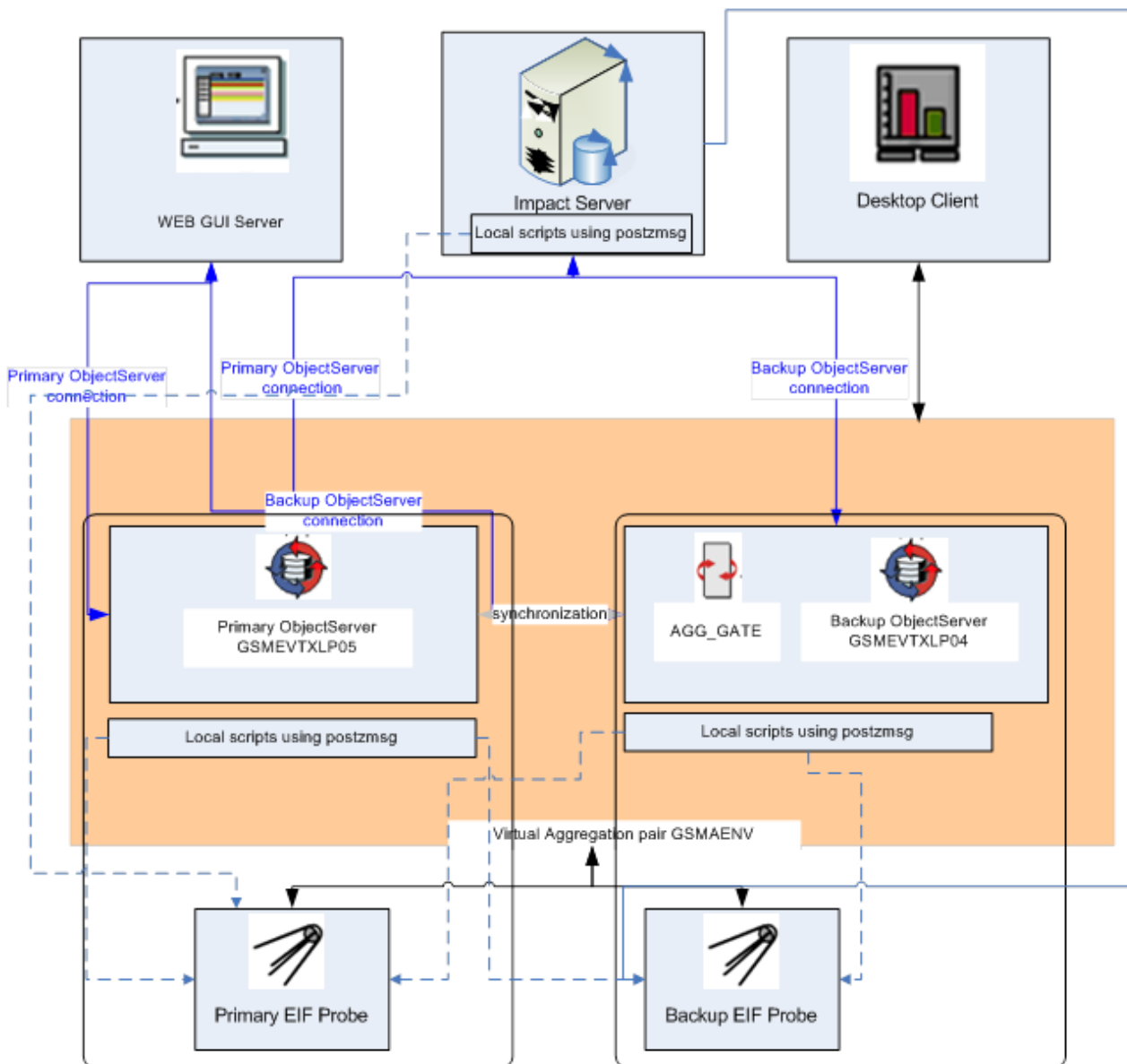


Figure 1 - Overall OMNibus failover configuration

In the above figure, no assumption is done about the location of the EIF probes; they can be located on a remote server (HTEMS, ...) as well as on the ObjectServer itself (it is this one which is used to support event update requests sent from the Impact server (using the command postzmsg) from automation scripts).

In the LIGHT configuration, when some automation scripts are moved from Impact to the ObjectServer itself, the postzmsg command is invoked locally to the local EIF probe running on the ObjectServer.

In a failover configuration, the postzmsg command must be redirected to the Virtual Aggregation pair (Vap), rather than to a specific ObjectServer (see 7, EIF probe Failover).

All connections to the Backup ObjectServer are periodically closed by backup process, so user cannot logon to this Object Server Event List.

In the figure, the aggregation pair of ObjectServers is connected by a bidirectional Object-Server Gateway to keep the ObjectServers synchronized, and the bidirectional ObjectServer Gateway runs on the backup host.

Probes connect directly to the virtual aggregation pair (GSMAENV) to facilitate failover and fail-back if the primary aggregation ObjectServer computer becomes unavailable. Alternative targets to which alerts can be forwarded from the aggregation layer are also shown:

Impact Server or WEBGUI accesses the aggregation failover pair using the primary and backup definition for ObjectServers. These products do not use the GSMAENV virtual server.

To reduce the time taken to resynchronize the contents of one ObjectServer with the another in the pair, after the recovery, the GSMA Failover solution uses the Gate.Resync.Type property to specify the type of resynchronization that is required. Set Gate.Resync.Type to "Minimal" to configure the gateway to resynchronize only events that were inserted or updated into the source ObjectServer after the other ObjectServer or the gateway failed.

To minimize event loss, which can occur if clients (probes) switch back to the primary ObjectServer before resynchronization is completed, client failback behaviour must be controlled by a failover pair of ObjectServers instead of the clients themselves.

1.2 Configuring server communication information

1.2.1 Processing

In normal situation, Primary and Backup Object Servers are running and synchronizing permanently.

WebGUI is connected to both primary and backup Object Servers. Impact is connected to Primary ObjectServer only. Probes are connected to the virtual aggregation pair, so probes are connected to the Primary ObjectServer.

Primary ObjectServer Properties are:

- ActingPrimary=true: the server is acting as the primary server, means automation are available.
- BackUpObjectServer=false : the server is the primary server.

Backup ObjectServer Properties are:

- ActingPrimary=false: the server is NOT acting as the primary server, means automation are disabled.
- BackUpObjectServer=true : the server is the backup server.

Every 1 minute, the backup Object Server disconnects all the eventual clients: PROBE, Impact, gateways etc... but WEBGUI and aggregation gateway.

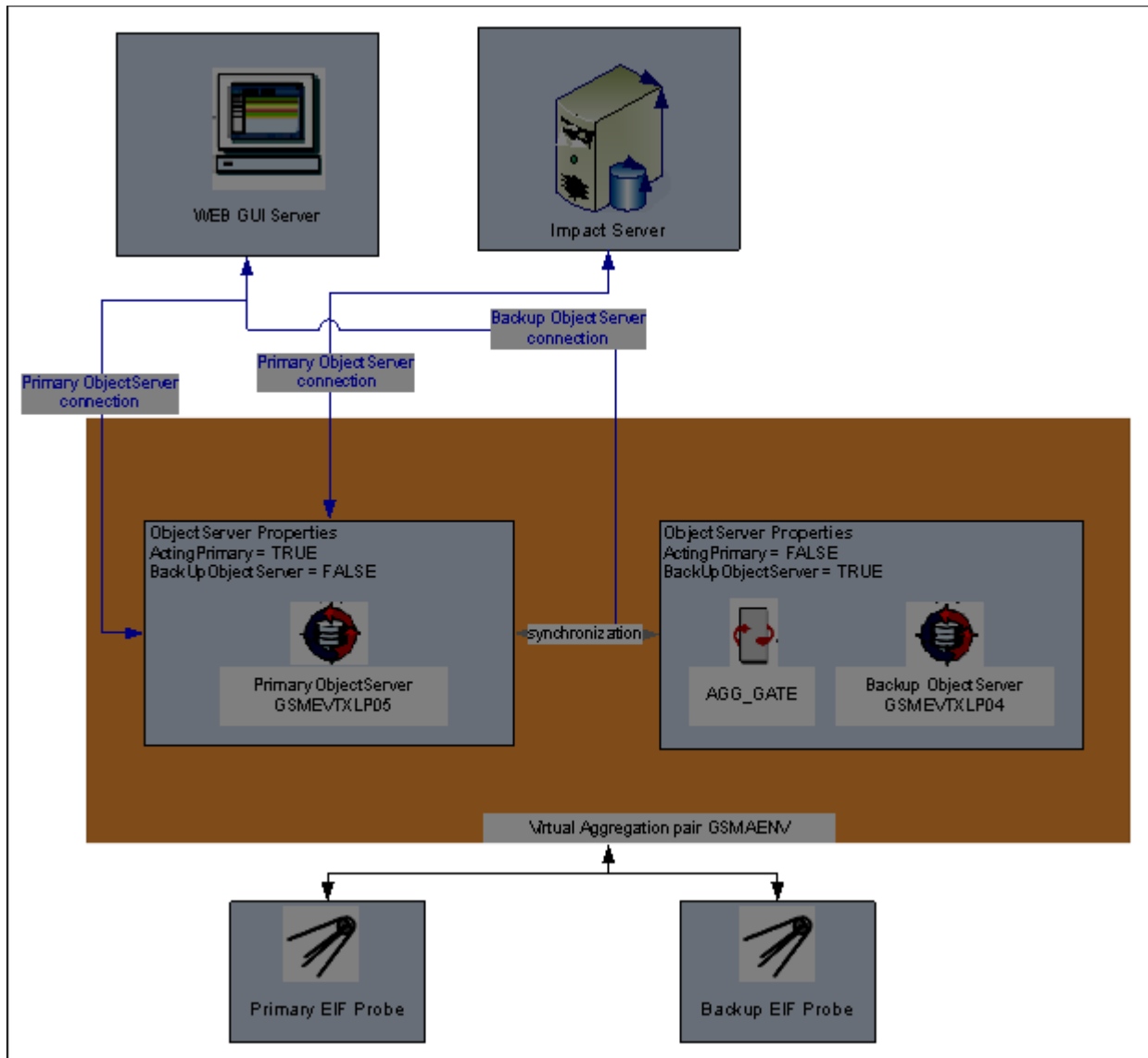


Figure 2 - ObjectServers communications during failover - Set-up

When Primary ObjectServer is down, WebGUI and Impact Server automatically switch to the Backup ObjectServer. They will ping periodically Primary ObjectServer in order to connect back to this server.

Probes are automatically routed to backup ObjectServer.

On Backup Object Server, properties are changed by the Aggregation gateway's signal (gw_counterpart_down) that the Primary ObjectServer is no more reachable.

ActingPrimary Property is set to true on Backup ObjectServer and, so, automation are enabled on the server. The backup Object Server do no more disconnects any clients.

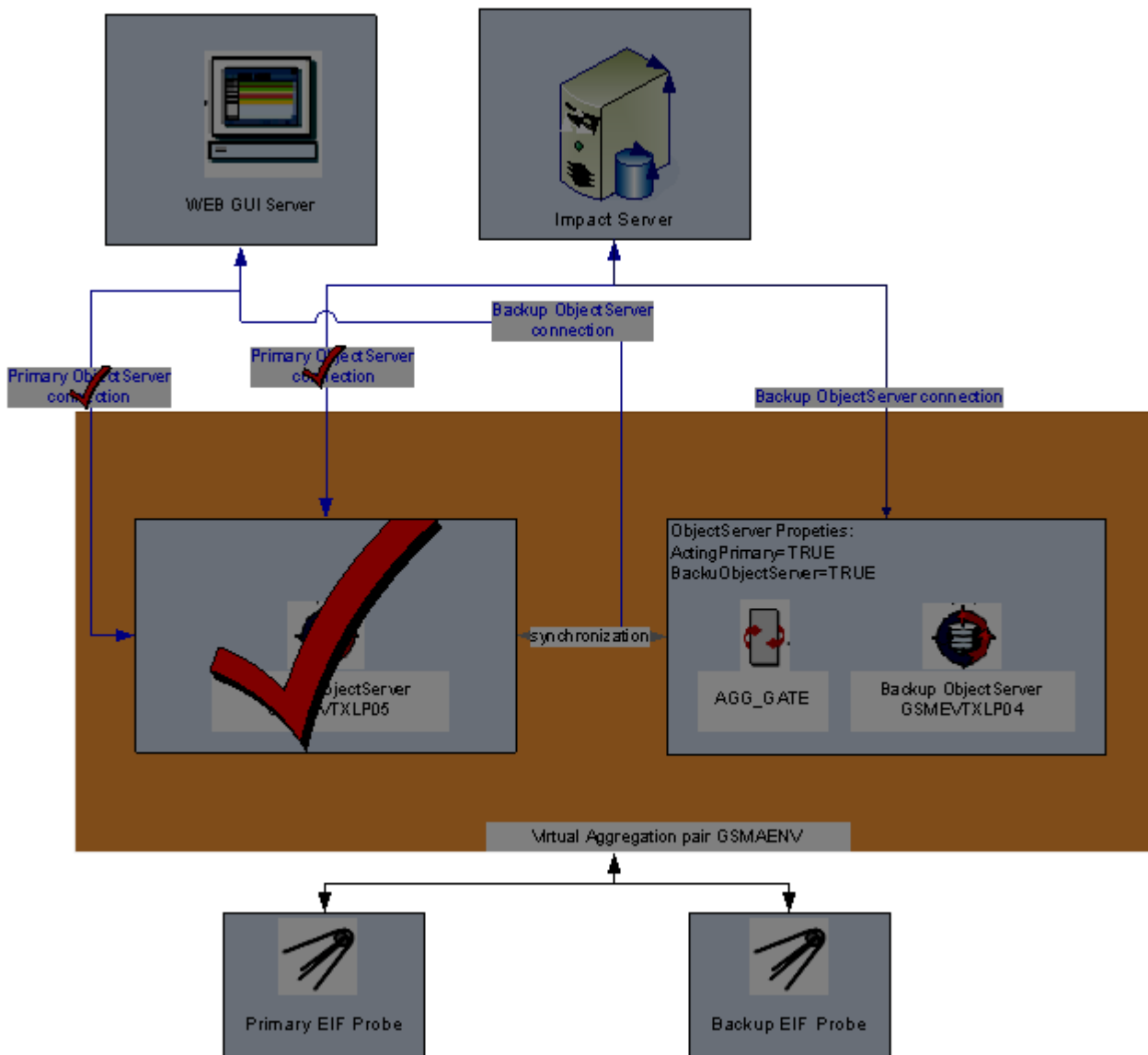


Figure 3 - ObjectServers communications during failover - Primary Up

When Primary Object Server is up, the aggregation gateway sends a signal (*gw_counterpart_up*) that the Primary ObjectServer is running.

ActingPrimary Property is set back to false on Backup ObjectServer and, so, automation is disabled on the server. The backup Object Server disconnects all the eventual clients: PROBE, Impact, gateways etc... but WEBGUI and aggregation gateway.

When failback is disabled for the clients, they remain connected to the backup ObjectServer until the backup ObjectServer forcefully disconnects them when resynchronization is completed.

To indicate that resynchronization is complete, the Aggregation Gateway sends a **gw_resync_finish** signal to both the primary and backup ObjectServers. On receipt of this signal, the backup ObjectServer disconnects the clients (*disconnect_all_clients* trigger) so that they can connect to the resynchronized primary ObjectServer.

To minimize event loss when an ObjectServer fails during resynchronization, an ObjectServer property **ActingPrimary** is used to define which ObjectServer was acting as the primary if the last resynchronization was not successful. The bidirectional ObjectServer Gateway determines

the direction of the resynchronization by using the ActingPrimary property of the ObjectServer. This property setting is updated solely by automations, and requires no user intervention.

1.2.2 Naming Convention

Three Object Server entities must be defined in server communication information for failover:

1. Primary Object Server (GSMEVTXLP04) must follow standard GSMA naming convention
2. Backup Object Server (GSMEVTXLP05) must follow standard GSMA naming convention
3. Virtual failover aggregation pair¹

1.2.3 Communication port setting

Each host on which (any) Netcool Omnibus components are running must be configured with server communication information that enables the components in the architecture to run and communicate with the other one.

Update the communication information for all the server (Omnibus) components in your deployment by manually editing the connections data file :

`$NCHOME/etc/omni.dat`

which is used to create the interfaces file.

A suggested good practice is, at Primary ObjectServer layer, to add all the components in the entire deployment to a single omni.dat file, which can then be distributed to all the hosts in the deployment. Only the backup ObjectServer will have a different host description for NCO_PA as we need a process agent running on both primary and backup ObjectServer. You can then generate the interfaces file from each computer by running the `$NCHOME/bin/nco_igen` command, as described in later procedures. (Interfaces files are named `$NCHOME/etc/interfaces.arch`, where *arch* is the operating system name.) Sample omni.dat files are provided to show the configuration for the servers in the aggregation layer.

Warning: all Object Servers involved in the Virtual failover aggregation pair must have been updated to the exact same GSMA ObjectServer version (whatever the version used).

Example: `gsmevtxlp05.lagaude.ibm.com` is the primary server, `gsmevtxlp04.lagaude.ibm.com` is the backup server and `GSMAENV` is the virtual failover gateway pair name.

```
#
# omni.dat file as prototype for interfaces file
#
# Ident: $Id: omni.dat 1.5 1999/07/13 09:34:20 chris Development $
#
[GSMEVTXLP05]
{
    Primary: gsmevtxlp05.lagaude.ibm.com 4100 ssl 4500
}
[GSMEVTXLP04]
{
    Primary: gsmevtxlp04.lagaude.ibm.com 4100 ssl 4500
}
[GSMAENV]
{
    Primary: gsmevtxlp05.lagaude.ibm.com 4100 ssl 4500
}
```

¹ Naming convention still be defined



```
Backup: gsmevtxlp04.lagaude.ibm.com 4100 ssl 4500
}
[AGG_GATE]
{
    Primary: gsmevtxlp04.lagaude.ibm.com 4900
}
[NCO_PA]
{
    Primary: gsmevtxlp05.lagaude.ibm.com 4200
}
```

On the backup the NCO_PA definition is:

```
[NCO_PA]
{
    Primary: gsmevtxlp04.lagaude.ibm.com 4200
}
```

2. Installing and Configuring the ObjectServers

2.1 Installing and configuring the Primary ObjectServer

Install the Netcool Products using the GSMA recommendation and GSMA packages. Ensure that all components are selected for installation (gateway in particular).

Please Refer to Omnibus Installation cookbook (complete installation).

Please see in 82, "List of default OMNibus triggers used during failover" the list of triggers involved in the ObjectServer failover mechanisms that do not have to be altered.

2.2 Installing and configuring the Backup ObjectServer

2.2.1 Install the Backup ObjectServer

Install the Netcool Products using the GSMA recommendation and GSMA package. Ensure that all components are selected for installation (including the gateway).

Please Refer to [GSMA ObjectServer Installation cookbook](#) Sections 3 to 5:

- Object Server intallation and configuration
- SUF (Situation Update Forwarder)

2.2.2 Configure the Backup ObjectServer

2.2.2.1 Using manual GSMA procedure

Install the GSMA triggers and procedures like for a standalone server (ObjectServer installation cookbook Section 4)

Once the server installed, it is important to check the order of fields defined in alert.status is exactly the same from the primary server to the backup server. If it is not the case, rebuild the alert.status table **uniquely** using the procedure described in next chapter. The PRIMARY_LIST must only contain:

ObjectServer GSMEVTXLP05 Table alerts.status

2.2.2.2 Using confpack export facility

This method would not drop or disable standard Tivoli triggers as recommended by GSMA installation.

- 1) From the Primary aggregation server, using ncosys user, extract the list of the Object Server components:

```
mkdir /opt/IBM/GSMA/ObjectServer/tmp
```

```
$OMNIHOME/bin/nco_confpack -list -user 'root' -password '<root password>' -server <Primary Object Server Name> -file /opt/IBM/GSMA/ObjectServer/tmp/PRIMARY_LIST
```


where <root password> is the password for root user but you can use ncosys user as well.

2) Configure the list of components to export

3) Edit the /opt/IBM/GSMA/ObjectServer/tmp/PRIMARY_LIST file:

The following list of components must be **removed** from the PRIMARY_LIST file (if exists). The list is not exhaustive, it contains all the triggers and SQLFile that only apply on one specific environment (for example the name of the SQLFILE contains the Object Server Name):

Type	name
Auto	aen_activity
Auto	block_clients_during_control_shutdown
Auto	client_activity_report
Auto	dedup_client_stats
Auto	dedup_status_inserts
Auto	details_inserts
Auto	final_shutdown
Auto	initialise_user_stats
Auto	itm_client_profile
Auto	itm_events_by_class
Auto	itm_events_by_node
Auto	itm_events_by_severity
Auto	itm_evtclass_add
Auto	itm_evtclass_del
Auto	itm_evtclass_update
Auto	itm_evtcount
Auto	itm_storesize
Auto	itm_tablesize
Auto	itm_trigger_stats_report
Auto	journal_inserts
Auto	new_status_inserts
Auto	profiler_group_report
Auto	profiler_report
Auto	profiler_toggle
Auto	status_updates
Auto	trigger_stats_report
Procedure	control_shutdown
Procedure	disable_control_shutdown

Procedure	enable_control_shutdown
Procedure	ext_shutdown
Procedure	itm_evtclass_write
SQLFile	aenstats
SQLFile	class
SQLFile	clntprofile
SQLFile	event_by_severity
SQLFile	evtclassdist
SQLFile	evtcount
SQLFile	evtnodedist
SQLFile	evtrate
SQLFile	profiler_report
SQLFile	storesize
SQLFile	tablesize
SQLFile	trigger_stats_report
SQLFile	trigstats

Table 1: Triggers to Remove

The following list of components must be **added** to the PRIMARY_LIST file (if not already included):

Type	name
Table	alerts.problem_events

Table 2: Triggers to Add

- 4) Export the Primary Object Server Configuraiton using nco_confpack command

```
$OMNIHOME/bin/nco_confpack -export -file
/opt/IBM/GSMA/ObjectServer/tmp/PRIMARY_LIST -rename <Primary Object Server
Name>:<Backup Object Server Name> -package /opt/IBM/GSMA/ObjectServer/tmp/PRI-
MARY.jar -user 'root' -password '<root password>'
```

where <Primary Object Server Name> is the Primary Object server name as defined in omni.dat file. In the example GSMEVTXLP05

where <Backup Object Server name> is the Backup Object Server name as defined in omni.dat file. In the example GSMEVTXLP04

where <root password> is the password for root as defined in the Object Server. You can use ncosys user as well.

- 5) Logon Backup Object Server using **ncosys** user.

- 6) Create the tmp directory on this object server

```
mkdir /opt/IBM/GSMA/ObjectServer/tmp
```

- 7) Copy the /opt/IBM/GSMA/ObjectServer/tmp/PRIMARY.jar file created in previous step in /opt/IBM/GSMA/ObjectServer/tmp directory on the backup object server.

8) Import the configuration if the Primary Object Server in the Backup Object Server.

```
$OMNIHOME/bin/nco_confpack -import -package /opt/IBM/GSMA/ObjectServer/tmp/PRIMA-
RY.jar -user 'root' -server <Backup Object Server Name> -password '<root password>'
-force
```

where <Backup Object Server name> is the Backup Object Server name as defined in omni.dat file.

In the example GSMEVTXLP04, where <root password> is the password for root as defined in the Object Server.

9) Stop and restart Object Server.

```
$OMNIHOME/bin/nco_pa_stop -process MasterObjectServer
$OMNIHOME/bin/nco_pa_start -process MasterObjectServer
$OMNIHOME/bin/nco_pa_status
```

10) Verify the standard Netcool triggers are not enabled if they exist:

- deduplication
- generic_clear
- delete_clears

1) Verify ITM6 triggers are not enabled if they exist:

- itm_event_send
- deleteitm
- itm_deduplication
- synchronizeitm
- itmerror
- itm_event_clear
- get_sit_timeout
- itm_event_clear

3.Failover configuration

Attention, the failover installation adds columns in alerts.status Omnibus table. In some Omnibus versions, adding new columns in alerts.status table fires a compilation error on all Omnibus triggers using this table. As a result these triggers are dropped. This is a known Omnibus error that must be corrected in March 2011.

So after the failover installation, check the Netcool log file and, if necessary re create dropped triggers.

3.1 Configuring Failover on the primary ObjectServer

The ObjectServer must have been installed previously according to GSMA standards (recommendations, installation packages).

Use the following steps to configure the Primary ObjectServer <Primary Object Server name> (GSMEVTXLP05 in the example), and apply the SQL customization.

To install and configure the ObjectServer:

1. Install Tivoli Netcool/OMNibus following complete GSMA recommendations and procedure. Ensure that all components are selected for installation (gateway will be installed on backup ObjectServer so if your environment do not need a gateway, it is not necessary to select this component).
2. Ensure that the \$NCHOME/etc/omni.dat is configured with all component details from primary and backup environments (see Configuring server communication information chapter and example).
3. Generate the interfaces file as follows:
`$NCHOME/bin/nco_igen`
4. Logon to the ObjectServer with **ncosys** ID.
5. Configure RPM Repository referring omnibus cookbook section 5.2
6. Download the gsma_NOS_failover_x.x.x.rpm package from <http://gsma.lagaude.ibm.-com/packages>
7. Place the gsma_NOS_failover_x.x.x.rpm file in the /tmp directory.
8. Edit and correct the /opt/IBM/GSMA/config/gsma_NOS_failover_automated_upgrade.silent file with :

1. the ObjectServer Name :

```
$main::NCO_name = "NCOMS";
```

2. the user used to administrate the ObjectServer (default is ncosys) :

```
$main::NCO_user = "ncosys";
```

3. the SQL password of the previous user (the password defined insode the Object-Server not the system password) :

```
$main::NCO_password = "<your_password>";
```

Note: According to new IBM password security policy, please ensure the passwords length is set with longer than 15 (>=15) for all user IDs. When you generate the 15 character's password by manual or password management tool, please exclude some specific characters (e.g., ""\>^\$&([,],{,},<,>,(,))

9. Install the rpm with command

```
rpm -dbpath /opt/IBM/GSMA/rpmdb -ivh /tmp/gsma_NOS_failover_<version>.rpm --ignoreos
```

10. The following command will run automatically:

```
$OMNIHOME/bin/nco_sql -user ncosys -server <objectservername> < /opt/IBM/GSMA/failover/install/gsma_aggregation.sql
$OMNIHOME/bin/nco_sql -user ncosys -server <objectservername> < /opt/IBM/GSMA/failover/install/gsma_aggregation_primary.sql
```

where objectservername is the name of the primary Object Server (GSMEVTXLP05 in the example).

The SQL customization is applied. This ObjectServer is defined as the primary Object-Server (BackupObjectServer set to FALSE ActingPrimary set to TRUE).

You could get an error message concerning an Object not found in logs: disconnect_all_clients. Do not take this message into account.

Note: For redhat linux, above the command should be changed into:

```
$OMNIHOME/bin/nco_sql -user ncosys -password <objectserver password> -server <objectservername> < /opt/IBM/GSMA/failover/install/gsma_aggregation.sql
$OMNIHOME/bin/nco_sql -user ncosys -password <objectserver password> -server <objectservername> < /opt/IBM/GSMA/failover/install/gsma_aggregation_primary.sql
```

11. Stop and restart Object Server.

```
$OMNIHOME/bin/nco_pa_stop -process MasterObjectServer
$OMNIHOME/bin/nco_pa_start -process MasterObjectServer
$OMNIHOME/bin/nco_pa_status
```

The ObjectServer is confirmed as initialized and entering a RUN state.

3.2 Configuring controlled shutdown of the Primary Object-Server

You can configure a controlled shutdown of any ObjectServer such that pending changes are forwarded to IDUC clients before the ObjectServer shuts down. This minimizes the possibility of data loss on shutdown.

To enable controlled shutdown, the ObjectServer schema must be updated with a set of triggers and procedures that are provided in an SQL import file gsma_control_shutdown.sql, which is stored in the /opt/IBM/GSMA/failover/install directory. The ObjectServer must also be set up to run under process control because the nco_pa_stop utility needs to be called from an external procedure to shut down the ObjectServer.

The triggers and procedures that are provided in the gsma_control_shutdown.sql file will orchestrate a controlled shutdown. The ObjectServer is first brought to a restricted state. Connections identified for non-IDUC clients (such as nco_sql and nco_config) are dropped, and an IDUC FLUSH command is initiated to send pending changes to all identified IDUC clients (such as gateways and event lists). Any new connection requests to the ObjectServer are blocked. If store and forward is enabled for probes, any new alerts are stored in a store-and-forward file until the probe can successfully reconnect to an ObjectServer. When the data retrieval is completed for the IDUC clients, the nco_pa_stop utility is used to shut down the ObjectServer process that is running under process control.

Sections of the `gsma_control_shutdown.sql` file must be edited to specify information that is required for setting up the configuration.

To configure and perform a controlled shutdown for an ObjectServer:

1. Logon `ncosys` user on the Primary ObjectServer
2. If your ObjectServer process name in Process Agent configuration file is not `MasterObjectServer`, correct the `gsma_control_shutdown.sql` file

on line : `execute procedure ext_shutdown ('MasterObjectServer', username, pass, paserver);`

Replace the `MasterObjectServer` default setting with the ObjectServer process name defined in the process agent configuration file.

3. Apply the controlled shutdown customization to a new or existing ObjectServer as follows:

```
$OMNIHOME/bin/nco_sql -user ncosys -server $objsrvr <
/opt/IBM/GSMA/failover/install/gsma_control_shutdown.sql
```

Note: For redhat linux, above the command should be changed into:

```
$OMNIHOME/bin/nco_sql -user ncosys -password <objectserver password> -server $objsrvr <
/opt/IBM/GSMA/failover/install/gsma_control_shutdown.sql
```

Assuming clients (both IDUC and non-IDUC) have been started within your environment, you can perform a controlled shutdown at any time by running the following SQL commands from the SQL interactive interface:

```
logon ncosys
```

```
/opt/IBM/GSMA/failover/scripts/gsma_Shutdown_ObjectServer.sh <process agent name>
<operating system password> <ncosys password>
```

where `<process agent name>` is the name of the process agent as defined in the `$NCHOME/etc/omni.dat` file (default `NCO_PA`).

`<operating system password>` is the password of `ncosys` in OS and `<ncosys password>` is the password of the `ncosys` in objectserver.

3.3 Configure Failover on the Backup Object Server

Use the following steps to configure the Backup ObjectServer `<Backup Object Server name>` (GSMEVTXLP04 in the example), and apply the SQL customization.

To configure the ObjectServer:

1. Logon to the Backup ObjectServer with **ncosys** ID.
2. Ensure that the `$NCHOME/etc/omni.dat` is configured with all component details from primary and backup environments (see **To minimize event loss, which can occur if clients (probes) switch back to the primary ObjectServer before resynchronization is completed, client fallback behaviour must be controlled by a failover pair of ObjectServers instead of the clients themselves.** chapter and example). Verify the `NCO_PA` definition : hostname must be the hostname of the backup ObjectServer.
3. Generate the interfaces file as follows:

```
$NCHOME/bin/nco_igen
```
4. Download the `gsma_NOS_failover_x.x.x.rpm` package from <http://gsma.lagaude.ibm.com/packages>

5. Place the gsma_NOS_failover_x.x.x.rpm file in the /tmp directory.
6. Install the rpm with command


```
rpm -dbpath /opt/IBM/GSMA/rpmdb -ivh /tmp/gsma_NOS_failover_<version>.rpm --ignoreos
```
7. the rpm installation script will Initialize the Backup ObjectServer (GSMEVTXLP04 in the example) and include the SQL import file to be applied to this ObjectServer:


```
$OMNIHOME/bin/nco_sql -user ncosys -server <objectservername> < /opt/IBM/GSMA/failover/install/gsma_aggregation.sql
$OMNIHOME/bin/nco_sql -user ncosys -server <objectservername> < /opt/IBM/GSMA/failover/install/gsma_aggregation_failover.sql
```

Where objectservername is the name of your Backup Object Server (GSMEVTXLP04 in the example).

If you have run nco_confpack procedure to configure ObjectServer, do not take into account messages for existing objects. You should get an error message concerning an Object not found: disconnect_all_clients. Do not take this message into account.

The SQL customization is applied. This ObjectServer is defined as the backup ObjectServer (BackupObjectServer set to TRUE, ActingPrimary set to FALSE, PrimaryOnly group automation triggers are disabled).

Note: For redhat linux, above the command should be changed into:

```
$OMNIHOME/bin/nco_sql -user ncosys -password <objectserver password> -server <objectservername> < /opt/IBM/GSMA/failover/install/gsma_aggregation.sql
$OMNIHOME/bin/nco_sql -user ncosys -password <objectserver password> -server <objectservername> < /opt/IBM/GSMA/failover/install/gsma_aggregation_failover.sql
```

8. Stop and restart Object Server.

```
$OMNIHOME/bin/nco_pa_stop -process MasterObjectServer
$OMNIHOME/bin/nco_pa_start -process MasterObjectServer
$OMNIHOME/bin/nco_pa_status
```

The ObjectServer is confirmed as initialized and entering a RUN state.

3.4 Defining User ID and roles for Aggregation Gateway on Primary and Backup Object Servers

The aggregation gateway (GSMAENV) must authenticate with the ObjectServers using a userid and password. The userid must have the correct authority to be able to insert records in the required tables.

At the first start of the aggregation gateway, in order to copy all information from the primary to the backup objectserver without encounter any authorization issue, administrator must define **root** user to synchronize all information between both ObjectServers. Once the synchronization has taken place (usual run), then GSMA recommends defining a userid named "agggateway" to be used by the aggregation gateway uniquely on each ObjectServers (primary and backup) and limit the synchronization scope to minimal.

Object Server Group The following groups should be created on the Omnibus Server for GSMA failover deployments. Additional Groups & Roles shall be created based on business requirements.

Function	Omnibus Group Name	Group Roles
GSMA Group for Aggregation Gateway	AggGateway	SuperUser AlertsUser AutoAdmin CatalogUser ChannelAdmin DatabaseAdmin DesktopAdmin ISQLWrite SecurityAdmin ToolsAdmin

Object Server User ID: The "agggateway" userid must be defined in group AggGateway.

A password should be defined for the user using local ObjectServer authentication.

Note: The application userid "agggateway" is not a real operating system userid, it exists in the ObjectServer application only.

Follow this procedure to define the "agggateway" ID Bring up the interface by issuing:

```
$NCHOME/omnibus/bin/nco_sql -server $objectservername -user root
-password <password>
```

1. At the sql prompt, enter the following commands to define the users and add them to the appropriate groups.

```
create group 'AggGateway' comment 'GSMA Group for Aggregation Gate-
way';
go
grant role 'SuperUser' to group 'AggGateway';
grant role 'AlertsUser' to group 'AggGateway';
grant role 'AutoAdmin' to group 'AggGateway';
grant role 'CatalogUser' to group 'AggGateway';
grant role 'ChannelAdmin' to group 'AggGateway';
grant role 'DatabaseAdmin' to group 'AggGateway';
grant role 'DesktopAdmin' to group 'AggGateway';
grant role 'ISQLWrite' to group 'AggGateway';
grant role 'SecurityAdmin' to group 'AggGateway';
grant role 'ToolsAdmin' to group 'AggGateway';
go
create user 'agggateway' full name 'Aggregation gateway' PAM FALSE
go
alter group 'AggGateway' assign members 'agggateway'
go
```

2. Now enter vi at the sql prompt to open up a temporary file in the vi editor. Then copy the following commands into the editor and change the ????? password values to the password you want to assign to each id. When you are done updating the lines, enter :wq to exit the editor and have the command entered. You just have to type "go" and press enter to run them.


```
alter user 'agggateway' set password '??????';
```

Note: According to new IBM password security policy, please ensure the passwords length is set with longer than 15 (≥ 15) for all user IDs. When you generate the 15 characters's password by manual or password management tool, please exclude some specific characters (e.g, "'\>^\$&([,],{,},<, >, (,))

3. Exit the nco_sql interface

```
exit
```

4. Now verify that the "agggateway" userid works by using it. Bring up the interface using:

```
$NCHOME/omnibus/bin/nco_sql -server $objectservername -user  
agggateway
```

Then enter the password to complete the login.

5. Exit the nco_sql interface

```
exit
```

3.5 Configuring the Aggregation Gateway on Backup Object Server

Use the following steps to configure the bidirectional aggregation ObjectServer Gateway AGG_GATE. Installation of Tivoli Netcool/OMNibus on Backup Object Server has already been preformed in previous step..

To configure the gateway:

- 1) Copy the property files for the gateway, to the default location where configuration and properties files are held:

```
cp /opt/IBM/GSMA/failover/etc/AGG_GATE.props $NCHOME/omnibus/etc/.
```

The following files defined in the gateway property file are located in to /opt/IBM/GSMA/failover/etc:

- o AGG_GATE.map
- o AGG_GATE.tblrep.def

- 1) Obtain the Object Server password for the 'root' userid and encrypt it using AES_FIPS encryption. Refer to section Setting up Property File Encryption for instructions on creating the encryption key file if it has not already been done on this server.

Encrypt by issuing:

```
$OMNIHOME/bin/nco_aes_crypt -c AES_FIPS -k $NCHOME/etc/security/keys/en-  
cryption.key <password>
```

where password is the Object Server password for the 'root' userid.

- 2) Edit the \$NCHOME/omnibus/etc/AGG_GATE.props file with the following lines:

```
Gate.Resync.Type           : 'NORMAL'  
Gate.ObjectServerA.Server  : '<Primary Object Server Name>'  
Gate.ObjectServerB.Server  : '<Backup Object Server Name'  
Gate.ObjectServerA.Username : 'root'  
Gate.ObjectServerA.Password : '<aes_fips_encrypted_password>'  
Gate.ObjectServerB.Username : 'root'
```

```
Gate.ObjectServerB.Password      : '<aes_fips_encrypted_password>'
Gate.PAAware                    : 1
Gate.PAAwareName                : 'NCO_PA'
```

Note you need to first use root user for complete synchronization between ObjectServers with Resynchronization type NORMAL.

- 3) If you have configured Aggregation pair to use SSL, add the following lines:

```
Gate.ObjectServerA.CommonNames   : 'GSMAENV'
Gate.ObjectServerB.CommonNames   : 'GSMAENV'
```

where GSMAENV is the name of the Virtual failover pair as defined in omni.dat file.

- 4) Start the gateway AGG_GATE:

```
$NCHOME/omnibus/bin/nco_g_objserv_bi -propsfile
$NCHOME/omnibus/etc/AGG_GATE.props &
```

The gateway is confirmed as initialized and entering a RUN state.

- 5) Check the aggregation gateway log file /opt/IBM/GSMA/logs/ObjectServer/AGG_GATE.log. Track eventual errors. The log file must contain the following record: *Resync Manager: Successful resynchronisation complete.*

3.6 Configure Process Agent for Aggregation Gateway on Backup Object Server

If the backup object server has been installed according to **GSMA_Omnibus 8.1 Installation Cookbook** this steps should have been already performed **automatically** by the nos_pad.sh script.

The Process Agent is required in the GSMA solution. It is used to start the Aggregation gateway. With the GSMA configuration, the Aggregation gateway will be restarted automatically if it goes down.

- 1) Add new process

```
nco_process 'AggregationGateway'
{
    Command '$NCHOME/omnibus/bin/nco_g_objserv_bi -propsfile
$NCHOME/omnibus/etc/AGG_GATE.props' run as 'ncosys'
    Host      = '<Backup Object Server Name>'
    Managed   = True
    RestartMsg = '${NAME} running as ${EUID} has been restored on ${HOST}.'
    AlertMsg   = '${NAME} running as ${EUID} has died on ${HOST}.'
    RetryCount = 0
    ProcessType = PaPA_AWARE
}
```

- 2) Add the new process to the Core nco_service:

```
process 'AggregationGateway' NONE
```

- 3) Shutdown the process agent using ncosys ID.

```
$NCHOME/omnibus/bin/nco_pa_shutdown -user ncosys
login Password :
```

Connected To PA Server [NCO_PA] Shutdown Options:



Enablement and Exploitation of GSMA

- 1) Shutdown Server leaving managed processes running.
- 2) Shutdown Server and stop all managed processes.
- 3) Exit shutdown interface.

Select Option [1-3] 1

Shutdown PA and stop processes.

The command will ask you for the ncosys operating system password.

Start the process agent (using root user since running in -secure mode) :

```
$NCHOME/omnibus/bin/nco_pad -secure
```

Netcool/OMNIBus Process Agent Daemon - Version 7.3.0

Netcool/OMNIBus PA API Library Version 7.3.0
Sybase Server-Library Release: 15.0

Server Settings :

```
Name of server           : NCO_PA
Path of used log file    :
/opt/IBM/tivoli/netcool/omnibus/log/NCO_PA.log
Configuration File      :
/opt/IBM/tivoli/netcool/omnibus/etc/nco_pa.conf
Child Output File       : /dev/null
Maximum logfile size    : 1024
Thread stack size       : 69632
Message Pool size       : 45568
PID Message Pool size   : 50
Rogue Process Timeout   : 30
Truncate Log            : False
Instantiate server to daemon : True
Internal API Checking    : False
No Configuration File    : False
Start Auto-start services : True
Authentication System    : UNIX
Trace Net library        : False
Trace message queues     : False
Trace event queues       : False
Trace TDS packets        : False
Trace mutex locks        : False
Host DNS name            : pokdevlab75
PID file (from $OMNIHOME) : ./var/nco_pa.pid
Kill Process group       : False
Secure Mode              : True
Administration Group Name. : ncoadmin
```

Forking to a Daemon Process.....

Note: if there are issues on PAM authentication when you start the PA:

nco_pa_status

Login Password:

2017-09-19T03:18:54: Error: Failed to make a connection to NCO_PA.

NCO_PA.log:

2017-09-19T03:18:54: Error: OS PAM authentication failed. [Authentication fail-

Document: GSMA_OMNibus_Failover_Cookbook.odt
Owner: Florence Rouanet-Rose

Date: 2019/10/30
Status: FINAL

Subject: Failover NOS Installation Cookbook

Page 27 of 82

ure].

Add the following file on the objectserver:

```
cat /etc/pam.d/netcool
#%PAM-1.0
auth include system-auth
account include system-auth
password substack system-auth
-password optional pam_gnome_keyring.so
```

3.7 Reconfigure Aggregation gateway for agggateway user

- 1) Obtain the Object Server password for the 'agggateway' userid and encrypt it using AES_FIPS encryption. Refer to section Setting up Property File Encryption for instructions on creating the encryption key file if it has not already been done on this server.

Encrypt by issuing:

```
$OMNIHOME/bin/nco_aes_crypt -c AES_FIPS -k $NCHOME/etc/security/keys/encryption.key
<password>
```

where password is the Object Server password for the 'agggateway' userid.

- 2) Edit the \$NCHOME/omnibus/etc/AGG_GATE.props file with the following lines:

```
Gate.Resync.Preferred           : 'ObjectServerA'
Gate.Resync.Type                : 'MINIMAL'
Gate.ObjectServerA.Username     : 'agggateway'
Gate.ObjectServerA.Password     : '<aes_fips_encrypted_password>'
Gate.ObjectServerB.Username     : 'agggateway'
Gate.ObjectServerB.Password     : '<aes_fips_encrypted_password>'
Gate.ObjectServerA.TblReplicateDefFile:
'/opt/IBM/GSMA/failover/etc/gsma_AGG_GATE.tblrep.def'
Gate.ObjectServerB.TblReplicateDefFile:
'/opt/IBM/GSMA/failover/etc/gsma_AGG_GATE.tblrep.def'
```

It is very important to change both userid from root to agggateway and Gate.Resync.Type to 'MINIMAL'.

The Gate.Resync.Type must be changed to 'MINIMAL' in order to to resynchronize only events that were inserted or updated into the source ObjectServer after the other ObjectServer or the gateway failed. If the agggateway user is used with "NORMAL" value in Resync.Type parameter, all users will be removed from the backup ObjectServer table. It is also important to modify values for Gate.ObjectServerA.TblReplicateDefFile and Gate.ObjectServerB.TblReplicateDefFile **from AGG_GATE.props to /opt/IBM/GSMA/failover/etc/gsma_AGG_GATE.tblrep.def** file.

- 3) Stop the gateway AGG_GATE:

```
$NCHOME/omnibus/bin/nco_pa_stop -process AggregationGateway
```

- 4) Check the aggregation gateway is stopped.

```
$NCHOME/omnibus/bin/nco_pa_status
```

- 5) Restart the aggregation gateway.

```
$NCHOME/omnibus/bin/nco_pa_start -process AggregationGateway
```

- 6) Check the aggregation gateway is started

```
$NCHOME/omnibus/bin/nco_pa_status
```

- 7) Check the aggregation gateway log file `/opt/IBM/GSMA/logs/ObjectServer/AGG_GATE.log`. Track eventual errors. The log file must contain the following record: *Resync Manager: Successful resynchronisation complete*.
- 8) **Important notice . Because Gate.Resync.Prefered parameters is used** , after ObjectServerA failover and fail back . Omnibus will check which objectserver is set to be preferred, but if ObjectServerB (backup) will be have a longest up time. Resynchronization direction will be set from ObjectserverB (backup) to ObjectServerA (primary) in order to against to act of such a situation, when ObjectServerA is up and working correctly , please restart ObjectServerB (backup) . In this situation this ObjectServerA will be longest acting and Resync direction will be always from ObjectServerA to ObjectServerB.

4. Secure SSL Communication setting

If the primary and backup object server have been installed according to **GSMA_Omnibus 8.1 Installation Cookbook** these steps should have been already performed **automatically** by the `nos_ssl.sh` script.

4.1 Primary Object Server configuration for SSL

If Primary Object Server has already been configured to use SSL, directly switch to chapter.

4.1.1 Create Certificate Key Database

This is a common step for all Netcool/Omnibus components (Object Servers, probes)

- 1) Login to ncosys ID to perform this step.
- 2) Create the certificate key database file using ikeyman.

The command line version is `$NCHOME/bin/nc_gskcmd`. The key database must be called `omni.kdb` and located in the `$NCHOME/etc/security/keys` directory for UNIX (Object-Server, probe) or `%NCHOME%\ini\security\keys` on WINDOWS (Client Desktop)

On UNIX, create the key database using:

```
$NCHOME/bin/nc_gskcmd -keydb -create -type cms -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -expire 7300 -stash
```

Note: According to new IBM password security policy, please ensure the passwords length is set with longer than 15 (≥ 15) for all user IDs. When you generate the 15 character's password by manual or password management tool, please exclude some specific characters (e.g, ""\>^\$&([,],{,},<,>,(,))

This will create the key database using the assigned password which will expire in 20 years. The assigned password shall be anything and has no relation to the ncosys password or any object server ID/passwords.

Note: Ensure that the assigned password is stored in secure department database or other sources for future use.

4.1.2 Create a CA certificate (on the ObjectServer)

The steps below must be followed if the ObjectServer is chosen as the CA Certificate Authority (for an internal network):

- ITD GSMA shared deployments should use the Primary Object Server as the Certificate Authority (CA); the ObjectServer can either generate its own "self-signed" CA, or can use a "commercial" CA from a third party provider (example: VeriSign)
- It is recommended to follow the same for GSMA dedicated deployments, unless there is a customer requirement to use customer specific CAs

The steps below are not required if a certificate is imported from a commercial CA.

- 1) Create self signed certificate on object server using:

```
$NCHOME/bin/nc_gskcmd -cert -create -db $NCHOME/etc/security/keys/omni.kdb -pw
<your_assigned_pw> -dn "CN=<descriptive name for CA>,O=IBM,OU=GSMA" -label <de-
scriptive name for CA> -default_cert no -expire 7300 -ca true -size 2048
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db in key DB section.

<descriptive name for CA> is the name of the CA certificate. We recommend using the following standard: GSMA_<Primary Object Server Short Hostname>_CA

- 2) Extract this CA Certificate to a file:

```
$NCHOME/bin/nc_gskcmd -cert -extract -db $NCHOME/etc/security/keys/omni.kdb -pw
<your_assigned_pw> -label <descriptive name for CA> -target
$NCHOME/etc/security/keys/<descriptive name for CA>.arm
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<descriptive name for CA> is the name of the CA certificate defined previously.

<descriptive name for CA>.arm is the file name and path of the extracted certificate. If the path is not specified, the file will be created in your current path.

Note: It is recommended to store the arm file under \$NCHOME/etc/security/keys

- 3) Distribute this self-signed CA certificate file <descriptive name for CA>.arm to all clients that will be connecting via SSL by transferring it manually to each client.

Note: The certificates need to be distributed to any Administrator Client, and Probes, Web GUI and Impact Server which are in not in the same segment as the Object Server.

4.1.3 Setting up SSL on Primary Object Server

- 1) Edit \$NCHOME/etc/omni.dat properties file:

You must update the omni.dat file as follows:

```
[<PrimaryObjectservername>]
{
    Primary:          <primary server hostname> 4100 ssl 4500
}

[GSMAENV]
{
    Primary: <primary server hostname> 4100 ssl 4500
    Backup:  <backup server hostname> 4100 ssl 4500
}
```

Where:

<PrimaryObjectservername> is the name for this Primary ObjectServer following the standard naming convention.

<primary server hostname> is the hostname of the Primary Object Server

4100 is the non-SSL port for Object Server communications

4500 is the SSL port for Object Server communications

- 2) Once this file is changed, you must generate the interfaces file by issuing the command:

```
$NCHOME/bin/nco_igen
```

- 3) If your server was previously configured to use SSL, remove the existing signing request:

```
$NCHOME/bin/nc_gskcmd -certreq -delete -db $NCHOME/etc/security/keys/omni.kdb -pw  
<your assigned pw> -label <PrimaryObjectservername>
```

- 4) (Re)Create the signing request for Primary Object Server.

```
$NCHOME/bin/nc_gskcmd -certreq -create -db $NCHOME/etc/security/keys/omni.kdb -pw  
<your assigned pw> -dn "CN=GSMAENV,O=IBM,OU=GSMA" -label <PrimaryObject-  
servername> -file $NCHOME/etc/security/keys/<PrimaryObjectservername>_req.arm  
-size 2048
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<**PrimaryObjectservername**> is the name of the Primary ObjectServer as defined in the omni.dat file

Note: Common Name (CN) must be the virtual failover gateway pair name as defined in omni.dat file. (GSMAENV)

- 5) If there is an independent CA, for dedicated implementations, send the noth <objectserver-name>_req.arm request file to the CA and ask CA to sign the file that it must send back to you.

- 6) If the ObjectServer is also the CA, sign this request:

```
$NCHOME/bin/nc_gskcmd -cert -sign -db $NCHOME/etc/security/keys/omni.kdb -pw <your  
assigned pw> -label GSMA_<Primary Object Server Short Hostname>_CA -file  
$NCHOME/etc/security/keys/<PrimaryObjectservername>_req.arm -target  
$NCHOME/etc/security/keys/<PrimaryObjectservernam>.arm -expire 7200
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<Primary Object Server Short Hostname> is the short hostname of the Primary Object Server and GSMA_<Primary Object Server Short Hostname>_CA is the name of the CA certificate get from Primary ObjectServer.

<**PrimaryObjectservername**> is the name of the Primary ObjectServer as defined in the omni.dat file

- 7) Receive the signed server certificate

```
$NCHOME/bin/nc_gskcmd -cert -receive -db $NCHOME/etc/security/keys/omni.kdb -pw  
<your assigned pw> -file $NCHOME/etc/security/keys/<PrimaryObjectServer>.arm -de-  
fault_cert yes
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<**PrimaryObjectservername**> is the name of the Primary ObjectServer as defined in the omni.dat file

- 8) Restart the ObjectServer to pick up all the changes. Use:

```
$OMNIHOME/bin/nco_pa_stop -process MasterObjectServer
$OMNIHOME/bin/nco_pa_start -process MasterObjectServer
$OMNIHOME/bin/nco_pa_status
```

You should see that the ObjectServer is running again.

4.2 Backup Object Server configuration for SSL

4.2.1 Setting up SSL Certificate Database

If the session to the primary ObjectServer will be using SSL, you need to setup an SSL keystore database and import the root certificate for the Certificate Authority that issues the certificate for the Object Server.

The GSMA standard would be for the Object Server to have a self-signed certificate so you would need to import the CA certificate from the Object Server.

To setup the database and import the CA certificate:

- 1) Login to ncosys ID on Backup ObjectServer
- 2) Create the certificate key database file using ikeyman. The command line version is \$NCHOME/bin/nc_gskcmd. The key database must be called omni.kdb and located in the \$NCHOME/etc/security/keys directory. Create the key database by issuing:

```
$NCHOME/bin/nc_gskcmd -keydb -create -type cms -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -expire 7300 -stash
```

This will create the key database using the assigned password which will expire in 20 years. The assigned password can be anything and has no relation to the ncosys password or any object server ID/passwords.

Note: Ensure that the assigned password is stored in secure department database or other sources for future use.

4.2.2 CA certificate

- 1) If a commercial CA is being used, the CA certificate may already be in the new certificate database. If not, you will need to import it similar to how the self-signed certificate is imported below.

Obtain the root CA certificate from the **Primary Object Server** and place it in the \$NCHOME/etc/security/keys directory. The GSMA standard name for a self-signed certificate is GSMA_<Primary Object Server Short Hostname>_CA.arm

Import the CA certificate using command:

```
$NCHOME/bin/nc_gskcmd -cert -add -db $NCHOME/etc/security/keys/omni.kdb -pw
<your_assigned_pw> -label GSMA_<Primary Object Server Short Hostname>_CA -file
GSMA_<Primary Object Server Short Hostname>_CA.arm
```

You can list all the CA certificates in the db using:

```
$NCHOME/bin/nc_gskcmd -cert -list all -db $NCHOME/etc/security/keys/om-
ni.kdb -pw <your_assigned_pw>
```

4.2.3 Setting-up SSL on the ObjectServer

- 1) Edit \$NCHOME/etc/omni.dat properties file:

You must update the omni.dat file as follows:

```
[<BackupObjectservername>]
{
    Primary:          <backup server hostname> 4100 ssl 4500
}

[GSMAENV]
{
    Primary: <primary server hostname> 4100 ssl 4500
    Backup:  <backup server hostname> 4100 ssl 4500
}
```

Where:

<BackupObjectservername> is the name for this Backup ObjectServer following the standard naming convention.

<backup server hostname> is the hostname of the Backup Object Server

4100 is the non-SSL port for Object Server communications

4500 is the SSL port for Object Server communications

- 2) Once this file is changed, you must generate the interfaces file by issuing the command:

```
$NCHOME/bin/nco_igen
```

- 3) Now create a signing request on object server.

```
$NCHOME/bin/nc_gskcmd -certreq -create -db $NCHOME/etc/security/keys/omni.kdb -pw
<your_assigned_pw> -dn "CN=GSMAENV,O=IBM,OU=GSMA" -label <BackupObject-
servername> -file $NCHOME/etc/security/keys/<BackupObjectservername>_req.arm
-size 2048
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<BackupObjectservername> is the name of the Backup ObjectServer as defined in the omni.dat file above.

Note: Common Name (CN) must be set to the virtual failover gateway pair name as defined in omni.dat file. (GSMAENV)

- 4) If there is an independent CA, for dedicated implementations, send the **<BackupObjectservername>_req.arm** request file to the CA and ask CA to sign the file that it must send back to you.
- 5) If the ObjectServer is also the CA, send the **<BackupObjectservername>_req.arm** file to the Primary ObjectServer into `$NCHOME/etc/security/keys/` directory
- 6) Log on Primary ObjectServer, using `ncosys`.
- 7) From Primary ObjectServer, sign this request:

```
$NCHOME/bin/nc_gskcmd -cert -sign -db $NCHOME/etc/security/keys/omni.kdb -pw
<your assigned pw> -label GSMA_<Primary Object Server Short Hostname>_CA
-file $NCHOME/etc/security/keys/<BackupObjectservername>_req.arm -target
$NCHOME/etc/security/keys/<BackupObjectservername>.arm -expire 7200
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.
<Primary Object Server Short Hostname> is the short hostname of the Primary Object Server and **GSMA_<Primary Object Server Short Hostname>_CA** is the name of the CA certificate get from **Primary** ObjectServer.

<BackupObjectservername> is the name of the **Backup** ObjectServer as defined in the omni.dat file above.

Note: Use ObjectServer Name and NOT the hostname

- 8) Logon Backup ObjectServer, using `ncosys`
- 9) Receive the signed server certificate **<BackupObjectservername>.arm** in `$NCHOME/etc/security/keys` directory from Primary Object Server.

```
$NCHOME/bin/nc_gskcmd -cert -receive -db $NCHOME/etc/security/keys/omni.kdb -pw
<your assigned pw> -file $NCHOME/etc/security/keys/<BackupObjectserver-
name>.arm -default_cert yes
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<BackupObjectservername> is the name of the **Backup** ObjectServer as defined in the omni.dat file above.

- 10) Restart the ObjectServer to pick up all the changes. Use:

```
$OMNIHOME/bin/nco_pa_stop -process MasterObjectServer
$OMNIHOME/bin/nco_pa_start -process MasterObjectServer
$OMNIHOME/bin/nco_pa_status
```

You should see that the ObjectServer is running again.

5. Secure SSL Communication & FIPS setting

If the primary and backup object server have been installed according to **GSMA_Omnibus 8.1 Installation Cookbook** these steps should have been already performed **automatically** by the `nos_ssl.sh` and `nos_fips.sh` scripts.

5.1 Primary Object Server configuration for SSL & FIPS

If Primary Object Server has already been configured to use SSL, directly switch to chapter .

5.1.1 FIPS configuration

FIPS Configuration will be automatically enabled by NOS component installation however if it's not enabled during automated installation then below steps can be used for configuring the same.

Section 1 : A FIPS configuration file is required for FIPS initialization. This file is called `fips.conf`, and is required on each computer where a server component is installed.

Create an empty text file called `fips.conf` under `$NCHOME/etc/security/` folder if not exists.

touch \$NCHOME/etc/security/fips.conf

1. The encryption key file should be created using below command
\$OMNIHOME/bin/nco_keygen -o \$NCHOME/etc/security/keys/encryption.key'
2. The password for ncosys OS user should be encrypted using below command
\$NCHOME/omnibus/bin/nco_aes_crypt -c AES_FIPS -k \$NCHOME/etc/security/keys/encryption.key ' <GSMA_NCO_USERS_NCOSYS_PASS>'
3. The object server property file should be updated with encrypted cyptertext which is obtained as output from above command mentioned in line 3.
Also please uncomment and set the below mentioned parameters at object server to recreate object server user passwords.

ConfigCryptoAlg: 'AES_FIPS'
ConfigKeyFile: '\$NCHOME/etc/security/keys/encryption.key'
PA.Password: '<ENCR_PASS>'
PasswordEncryption: 'AES'

4. Passwords are recreated on the object server

5.1.2 Create Certificate Key Database

This is a common step for all Netcool/Omnibus components (Object Servers, probes)

- 1) Login to ncosys ID to perform this step.
- 2) Create the certificate key database file using `keyman`.

The command line version is `$NCHOME/bin/nc_gskcmd`. The key database must be called `omni.kdb` and located in the `$NCHOME/etc/security/keys` directory for UNIX (Object-Server, probe) or `%NCHOME%\ini\security\keys` on WINDOWS (Client Desktop)

On UNIX, create the key database using:

```
$NCHOME/bin/nc_gskcmd -keydb -create -type cms -db /opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb -pw <your_assigned_pw> -expire 7300 -stash -strong -fips
```

This will create the key database using the assigned password which will expire in 20 years. The assigned password shall be anything and has no relation to the ncosys password or any object server ID/passwords.

Note: Ensure that the assigned password is stored in secure department database or other sources for future use.

Note: According to new IBM password security policy, please ensure the passwords length is set with longer than 15 (≥ 15) for all user IDs. When you generate the 15 character's password by manual or password management tool, please exclude some specific characters (e.g, "\>^\$&[.,{,},<,>,(,))

5.1.3 Create a CA certificate (on the ObjectServer)

The steps below must be followed if the ObjectServer is chosen as the CA Certificate Authority (for an internal network):

- ITD GSMA shared deployments should use the Primary Object Server as the Certificate Authority (CA); the ObjectServer can either generate its own "self-signed" CA, or can use a "commercial" CA from a third party provider (example: VeriSign)
- It is recommended to follow the same for GSMA dedicated deployments, unless there is a customer requirement to use customer specific CAs

The steps below are not required if a certificate is imported from a commercial CA.

1) Create self signed certificate on object server using:

```
$NCHOME/bin/nc_gskcmd -cert -create -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -dn "CN=<descriptive name for CA>,O=IBM,OU=GSMA" -label <descriptive name for CA> -default_cert no -expire 7300 -ca true -size 2048
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db in key DB section.

<descriptive name for CA> is the name of the CA certificate. We recommend using the following standard: GSMA_<Primary Object Server Short Hostname>_CA

Example :

```
$NCHOME/bin/nc_gskcmd -cert -create -db $NCHOME/etc/security/keys/omni.kdb -pw tivoliNetCool@zxc -dn "CN=GSMA_gsmvtxlp08_CA,O=IBM,OU=GSMA" -label GSMA_gsmvtxlp08_CA -default_cert no -expire 7300 -ca true -size 2048
```

2) Extract this CA Certificate to a file:

```
$NCHOME/bin/nc_gskcmd -cert -extract -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -label <descriptive name for CA> -target $NCHOME/etc/security/keys/<descriptive name for CA>.arm -fips
```

Where:

<<your_assigned_pw> is the password you assigned to the omni.db key DB.

<descriptive name for CA> is the name of the CA certificate defined previously.

<descriptive name for CA>.arm is the file name and path of the extracted certificate. If the path is not specified, the file will be created in your current path.

Note: It is recommended to store the arm file under \$NCHOME/etc/security/keys

Example :

```
$NCHOME/bin/nc_gskcmd -cert -extract -db $NCHOME/etc/security/keys/omni.kdb -pw
tivoliNetCool@zxc -label GSMA_gsmvtxlp08_CA -target
$NCHOME/etc/security/keys/GSMA_gsmvtxlp08_CA.arm -fips
```

- 3) Distribute this self-signed CA certificate file <descriptive name for CA>.arm to all clients that will be connecting via SSL by transferring it manually to each client.

Note: The certificates need to be distributed to any Administrator Client, and Probes, Web GUI and Impact Server which are in not in the same segment as the Object Server.

5.1.4 Setting up SSL on Primary Object Server

- 1) Edit \$NCHOME/etc/omni.dat properties file:

You must update the omni.dat file as follows:

```
[<PrimaryObjectservername>]
{
    Primary:      <primary server hostname> 4100 ssl 4500
}

[GSMAENV]
{
    Primary: <primary server hostname> 4100 ssl 4500
    Backup: <backup server hostname> 4100 ssl 4500
}
```

Where:

<PrimaryObjectservername> is the name for this Primary ObjectServer following the standard naming convention.

<primary server hostname> is the hostname of the Primary Object Server

4100 is the non-SSL port for Object Server communications

4500 is the SSL port for Object Server communications

- 2) Once this file is changed, you must generate the interfaces file by issuing the command:

```
$NCHOME/bin/nc_igen
```

- 3) If you server was previously configured to use SSL, remove the existing signing request:

```
$NCHOME/bin/nc_gskcmd -certreq -delete -db $NCHOME/etc/security/keys/omni.kdb -pw
<your assigned pw> -label <PrimaryObjectservername>
```

- 4) (Re)Create the signing request for Primary Object Server.

```
$NCHOME/bin/nc_gskcmd -certreq -create -db $NCHOME/etc/security/keys/omni.kdb -pw <your as-
signed pw> -dn "CN=GSMAENV,O=IBM,OU=GSMA" -label <PrimaryObjectservername> -file
$NCHOME/etc/security/keys/<PrimaryObjectservername>_req.arm -fips -size 2048
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<**PrimaryObjectservername**> is the name of the Primary ObjectServer as defined in the omni.dat file

Note: Common Name (CN) must be the virtual failover gateway pair name as defined in omni.dat file. (GSMAENV)

Example :

```
$NCHOME/bin/nc_gskcmd -certreq -create -db
/opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb -pw ttivoliNetCool@zxc -dn 'CN=GS-
MAENV,O=IBM,OU=GSMA' -label XLP08 -file
/opt/IBM/tivoli/netcool/etc/security/keys/XLP08_req.arm -fips -size 2048
```

- 5) If there is an independent CA, for dedicated implementations, send the noth <objectserver-name>_req.arm request file to the CA and ask CA to sign the file that it must send back to you.
- 6) If the ObjectServer is also the CA, sign this request:

```
$NCHOME/bin/nc_gskcmd -cert -sign -db $NCHOME/etc/security/keys/omni.kdb -pw <your assigned
pw> -label GSMA_<Primary Object Server Short Hostname>_CA -file
$NCHOME/etc/security/keys/<PrimaryObjectservername>_req.arm -target
$NCHOME/etc/security/keys/<PrimaryObjectservernam>.arm -expire 7200 -fips
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<Primary Object Server Short Hostname> is the short hostname of the Primary Object Server and GSMA_<Primary Object Server Short Hostname>_CA is the name of the CA certificate get from Primary ObjectServer.

<**PrimaryObjectservername**> is the name of the Primary ObjectServer as defined in the omni.dat file

Example:

```
$NCHOME/bin/nc_gskcmd -cert -sign -db /opt/IBM/tivoli/netcool/etc/security/keys/om-
ni.kdb -pw tivoliNetCool@zxc -label GSMA_gsmevtxlp08_CA -file
/opt/IBM/tivoli/netcool/etc/security/keys/XLP08_req.arm -target
/opt/IBM/tivoli/netcool/etc/security/keys/XLP08.arm -expire 7200 -fips
```

- 7) Receive the signed server certificate

```
$NCHOME/bin/nc_gskcmd -cert -receive -db $NCHOME/etc/security/keys/omni.kdb -pw <your as-
signed pw> -file $NCHOME/etc/security/keys/<PrimaryObjectServer>.arm -default_cert yes -fips
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<**PrimaryObjectservername**> is the name of the Primary ObjectServer as defined in the omni.dat file

Example :

```
$NCHOME/bin/nc_gskcmd -cert -receive -db /opt/IBM/tivoli/netcool/etc/security/keys/om-
ni.kdb -pw tivoliNetCool@zxc -file /opt/IBM/tivoli/netcool/etc/security/keys/XLP08.arm -de-
fault_cert yes -fips
```

- 8) Restart the ObjectServer to pick up all the changes. Use:

```
$OMNIHOME/bin/nco_pa_stop -process MasterObjectServer
```



```
$OMNIHOME/bin/nc_o_pa_start -process MasterObjectServer
```

```
$OMNIHOME/bin/nc_o_pa_status
```

You should see that the ObjectServer is running again.

5.2 Backup Object Server configuration for SSL & FIPS

5.2.1 FIPS configuration

FIPS Configuration will be automatically enabled by NOS component installation however if it's not enabled during automated installation then below steps can be used for configuring the same.

Section 1: A FIPS configuration file is required for FIPS initialization. This file is called fips.conf, and is required on each computer where a server component is installed.

Create an empty text file called fips.conf under \$NCHOME/etc/security/ folder if not exists.

touch \$NCHOME/etc/security/fips.conf

- 1 The encryption key file should be created using below command
\$OMNIHOME/bin/nc_o_keygen -o \$NCHOME/etc/security/keys/encryption.key'
- 2 The password for ncosys OS user should be encrypted using below command
**\$NCHOME/omnibus/bin/nc_o_aes_crypt -c AES_FIPS -k
\$NCHOME/etc/security/keys/encryption.key '
<GSMA_NCO_USERS_NCOSYS_PASS>'**
- 3 The object server property file should be updated with encrypted ciphertext which is obtained as output from above command mentioned in line 3.
Also please uncomment and set the below mentioned parameters at object server to recreate object server user passwords.

ConfigCryptoAlg: 'AES_FIPS'

ConfigKeyFile: '\$NCHOME/etc/security/keys/encryption.key'

PA.Password: '<ENCR_PASS>'

PasswordEncryption: 'AES'

- 4 Passwords are recreated on the object server

5.2.2 Setting up SSL Certificate Database

If the session to the primary ObjectServer will be using SSL, you need to setup an SSL keystore database and import the root certificate for the Certificate Authority that issues the certificate for the Object Server.

The GSMA standard would be for the Object Server to have a self-signed certificate so you would need to import the CA certificate from the Object Server.

To setup the database and import the CA certificate:

- 1) Login to ncosys ID on Backup ObjectServer
- 2) Create the certificate key database file using ikeyman. The command line version is \$NCHOME/bin/nc_gskcmd. The key database must be called omni.kdb and located in the \$NCHOME/etc/security/keys directory. Create the key database by issuing:


```
$NCHOME/bin/nc_gskcmd -keydb -create -type cms -db /opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb -pw <your_assigned_pw> -expire 7300 -stash -strong -fips
```

This will create the key database using the assigned password which will expire in 20 years. The assigned password can be anything and has no relation to the ncosys password or any object server ID/passwords.

Note: Ensure that the assigned password is stored in secure department database or other sources for future use.

Note: According to new IBM password security policy, please ensure the passwords length is set with longer than 15 (≥ 15) for all user IDs. When you generate the 15 character's password by manual or password management tool, please exclude some specific characters (e.g, ""\>^\$&([,],{,},<,>,(,))

5.2.3 CA certificate

- 1) If a commercial CA is being used, the CA certificate may already be in the new certificate database. If not, you will need to import it similar to how the self-signed certificate is imported below.

Obtain the root CA certificate from the **Primary Object Server** and place it in the \$NCHOME/etc/security/keys directory. The GSMA standard name for a self-signed certificate is GSMA_<virtual pair name>_CA.arm

Import the CA certificate using command:

```
$NCHOME/bin/nc_gskcmd -cert -add -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -label GSMA_<Primary Object Server Short Hostname>_CA -file GSMA_<Primary Object Server Short Hostname>_CA.arm -fips
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<**PrimaryObjectservername**> is the name of the Primary ObjectServer as defined in the omni.dat file

Example :

```
$NCHOME/bin/nc_gskcmd -cert -add -db "$NCHOME/etc/security/keys/omni.kdb" -pw tivoliNetCool@zxc -label GSMA_gsmevtxlp08_CA -file /opt/IBM/tivoli/netcool/etc/security/keys/GSMA_gsmevtxlp08_CA.arm -fips
```

You can list all the CA certificates in the db using:

```
$NCHOME/bin/nc_gskcmd -cert -list all -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw>
```

5.2.4 Setting-up SSL on the ObjectServer

- 1) Edit \$NCHOME/etc/omni.dat properties file:

You must update the omni.dat file as follows:

```
[<BackupObjectservername>]
{
    Primary:      <backup server hostname> 4100 ssl 4500
}

[GSMAENV]
{
    Primary: <primary server hostname> 4100 ssl 4500
    Backup: <backup server hostname> 4100 ssl 4500
}
```

Where:

<BackupObjectservername> is the name for this Backup ObjectServer following the standard naming convention.

<backup server hostname> is the hostname of the Backup Object Server

4100 is the non-SSL port for Object Server communications

4500 is the SSL port for Object Server communications

- 2) Once this file is changed, you must generate the interfaces file by issuing the command:

```
$NCHOME/bin/nco_igen
```

- 3) Now create a signing request on object server.

```
$NCHOME/bin/nc_gskcmd -certreq -create -db
/opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb -pw <your assigned pw> -dn
'CN=GSMAENV,O=IBM,OU=GSMA' -label <BackupObjectservername> -file
/opt/IBM/tivoli/netcool/etc/security/keys/<BackupObjectservername>_req.arm -fips
-size 2048
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<BackupObjectservername> is the name of the Backup ObjectServer as defined in the omni.dat file above.

Note: Common Name (CN) must be set to the virtual failover gateway pair name as defined in omni.dat file. (GSMAENV)

Example:

```
$NCHOME/bin/nc_gskcmd -certreq -create -db
/opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb -pw tivoliNetCool@zxc -dn 'CN=GS-
MAENV,O=IBM,OU=GSMA' -label GSMEVTXLP04 -file
/opt/IBM/tivoli/netcool/etc/security/keys/GSMEVTXLP04_req.arm -fips -size 2048
```

- 4) If there is an independent CA, for dedicated implementations, send the <BackupObjectservername>_req.arm request file to the CA and ask CA to sign the file that it must send back to you.
- 5) If the ObjectServer is also the CA, send the <BackupObjectservername>_req.arm file to the Primary ObjectServer into \$NCHOME/etc/security/keys/ directory

6) Log on Primary ObjectServer, using ncosys.

7) From Primary ObjectServer, sign this request:

```
$NCHOME/bin/nc_gskcmd -cert -sign -db $NCHOME/etc/security/keys/omni.kdb -pw <your assigned pw> -label GSMA_<Primary Object Server Short Hostname>_CA -file
$NCHOME/etc/security/keys/<BackupObjectservername>_req.arm -target
$NCHOME/etc/security/keys/<BackupObjectservername>.arm -expire 7200 -fips
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.
 <Primary Object Server Short Hostname> is the short hostname of the Primary Object Server and **GSMA_<Primary Object Server Short Hostname>_CA** is the name of the CA certificate get from **Primary** ObjectServer.

<BackupObjectservername> is the name of the **Backup** ObjectServer as defined in the omni.dat file above.

Note: Use ObjectServer Name and NOT the hostname

Example :

```
$NCHOME/bin/nc_gskcmd -cert -sign -db /opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb -pw tivoliNetCool@zxc -label GSMA_gsmevtXlp08_CA -file
/opt/IBM/tivoli/netcool/etc/security/keys/GSMEVTXLP04_req.arm -target /opt/IBM/tivoli/netcool/etc/security/keys/GSMEVTXLP04.arm -expire 7200 -fips
```

8) Logon Backup ObjectServer, using ncosys

9) Copy the signed server certificate <BackupObjectservername>.arm in to \$NCHOME/etc/security/keys directory from Primary Object Server.

```
$NCHOME/bin/nc_gskcmd -cert -receive -db /opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb -pw <your assigned pw> -file /opt/IBM/tivoli/netcool/etc/security/keys/<BackupObjectservername>.arm -default_cert yes -fips
```

Where:

<your_assigned_pw> is the password you assigned to the omni.db key DB.

<BackupObjectservername> is the name of the **Backup** ObjectServer as defined in the omni.dat file above.

Example :

```
$NCHOME/bin/nc_gskcmd -cert -receive -db /opt/IBM/tivoli/netcool/etc/security/keys/omni.kdb -pw tivoliNetCool@zxc -file
/opt/IBM/tivoli/netcool/etc/security/keys/GSMEVTXLP04.arm -default_cert yes -fips
```

10) Restart the ObjectServer to pick up all the changes. Use:

```
$OMNIHOME/bin/nco_pa_stop -process MasterObjectServer
$OMNIHOME/bin/nco_pa_start -process MasterObjectServer
$OMNIHOME/bin/nco_pa_status
```

You should see that the ObjectServer is running again.

6. Configuration of the Virtual Aggregation Pair clients

6.1 Configure Probe to route events to the virtual aggregation pair

When probes are connecting to the virtual aggregation pair, disable failback in the probes by configuring them to connect to the aggregation (virtual) ObjectServer pair and by disabling polling. In the probe properties file:

- o Set Server to the aggregation Object Server pair name (GSMAENV in example).
- o Set ServerBackup to "".
- o Set PollServer to 0.

Make the changes following this process:

- 1) Logon **ncosys** user
- 2) Copy interfaces files from Primary ObjectServer \$NCHOME/etc/interfaces.arch, where arch is the operating system name

- 3) Update \$OMNIHOME/probes/aix5/tivoli_eif.props with the following lines:

```
Server      : '<Virtual Failover gateway pair name>'
ServerBackup : ''
```

Refer to the omni.dat definition for the *<Virtual Failover gateway pair name>*. In the example, the virtual gateway pair name is GSMAENV.

- 4) Edit the /opt/IBM/GSMA/probe/EIF/rules/gsmaTargets.include file and change the Object Server name in all target definitions statements (registertarget) to the name of the Virtual Failover Gateway pair.

Eg:- Alerts = registertarget("GSMAENV", "", "alerts.status")

- 5) Update \$OMNIHOME/probes/aix5/syntax.props with the following lines:

```
Server      : '< Virtual Failover gateway pair name >'
```

Refer to the omni.dat definition for the *<Virtual Failover gateway pair name>*. In the example, the virtual gateway pair name is GSMAENV.

- 6) Recycle the probe.

6.1.1 Setting up SSL Certificate Database

If the session to the ObjectServer will be using SSL, you need to setup an SSL keystore database and import the root certificate for the Certificate Authority that issues the certificate for the Object Server. Refer to section Configure the Object Server in secure SSL mode for information on when SSL is required.

If Probe witch will be configured is placed on Primary Objectserver or Backup Objectserver , please omit these chapter if steps from chapter 4 “Secure SSL Communication setting” were conducted.

The GSMA standard would be for the Object Server to have a self-signed certificate so you would need to import the CA certificate from the Object Sever.

To setup the database and import the CA certificate:

- 1) Login to ncosys ID.
- 2) Create the certificate key database file using ikeyman. The command line version is \$NCHOME/bin/nc_gskcmd. The key database must be called omni.kdb and located in the \$NCHOME/etc/security/keys directory. Create the key database by issuing:

```
$NCHOME/bin/nc_gskcmd -keydb -create -type cms -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -expire 7300 -stash
```

This will create the key database using the assigned password which will expire in 20 years. The assigned password can be anything and has no relation to the ncosys password or any object server ID/passwords.

Note: Ensure that the assigned password is stored in secure department database or other sources for future use.

Note: According to new IBM password security policy, please ensure the passwords length is set with longer than 15 (>=15) for all user IDs. When you generate the 15 character's password by manual or password management tool, please exclude some specific characters (e.g, ""\>^\$&([,],{,},<,>,(,))

- 3) If a commercial CA is being used, the CA certificate may already be in the new certificate database. If not, you will need to import it similar to how the self-signed certificate is imported below.

Obtain the root CA certificate from the Object Server and place it in the \$NCHOME/etc/security/keys directory. The GSMA standard name for a self-signed certificate is GSMA_<Object Server Short Hostname>_CA.arm

Import the CA certificate using command:

```
$NCHOME/bin/nc_gskcmd -cert -add -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -label GSMA_<Object Server Short Hostname>_CA -file GSMA_<Object Server Short Hostname>_CA.arm
```

You can list all the CA certificates in the db using:

```
$NCHOME/bin/nc_gskcmd -cert -list all -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw>
```

6.1.2 Setting up SSL & FIPS Certificate Database

If the session to the ObjectServer will be using SSL, you need to setup an SSL keystore database and import the root certificate for the Certificate Authority that issues the certificate for the Object Server. Refer to section Configure the Object Server in secure SSL mode for information on when SSL is required.

If Probe witch will be configured is placed on Primary Objectserver or Backup Objectserver , please omit these chapter if steps from chapter 4 “Secure SSL Communication setting” were conducted.

The GSMA standard would be for the Object Server to have a self-signed certificate so you would need to import the CA certificate from the Object Sever.

To setup the database and import the CA certificate:

- 1) Login to ncosys ID.
- 2) Create the certificate key database file using ikeyman. The command line version is \$NCHOME/bin/nc_gskcmd. The key database must be called omni.kdb and located in the \$NCHOME/etc/security/keys directory. Create the key database by issuing:

```
$NCHOME/bin/nc_gskcmd -keydb -create -type cms -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -expire 7300 -stash
```

This will create the key database using the assigned password which will expire in 20 years. The assigned password can be anything and has no relation to the ncosys password or any object server ID/passwords.

Note: Ensure that the assigned password is stored in secure department database or other sources for future use.

Note: According to new IBM password security policy, please ensure the passwords length is set with longer than 15 (≥ 15) for all user IDs. When you generate the 15 character's password by manual or password management tool, please exclude some specific characters (e.g, ""\>^\$&([,],{,},<,>,(,))

- 3) If a commercial CA is being used, the CA certificate may already be in the new certificate database. If not, you will need to import it similar to how the self-signed certificate is imported below.

Obtain the root CA certificate from the Object Server and place it in the \$NCHOME/etc/security/keys directory. The GSMA standard name for a self-signed certificate is GSMA_<Object Server Short Hostname>_CA.arm

Import the CA certificate using command:

```
$NCHOME/bin/nc_gskcmd -cert -add -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -label GSMA_<Object Server Short Hostname>_CA -file GSMA_<Object Server Short Hostname>_CA.arm -fips
```

Where:

<**your_assigned_pw**> is the password you assigned to the omni.db key DB.
 <**Primary Object Server Short Hostname**> is the short hostname of the Object Server and **GSMA_<Object Server Short Hostname>_CA** is the name of the CA certificate get from Primary ObjectServer which is signer.

Example :

```
$NCHOME/bin/nc_gskcmd -cert -add -db "$NCHOME/etc/security/keys/omni.kdb" -pw tivo-liNetCool@zxc -label GSMA_gsmevtxlp08_CA -file /opt/IBM/tivoli/netcool/etc/security/keys/GSMA_gsmevtxlp08_CA.arm -fips
```

You can list all the CA certificates in the db using:

```
$NCHOME/bin/nc_gskcmd -cert -list all -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw>
```

6.2 Configure WebGUI to connect on the virtual aggregation pair

Data Sources define the Object Servers that Web GUI server can communicate with. The object server parameters that are entered during Web GUI installation are stored in an xml file. The path and file name is shown below:

/opt/IBM/netcool/omnibus_webgui/etc/datasources/ncwDataSourceDefinitions.xml

The following lines define the Object Server details on the above xml file:

```
<!--
  ! Default datasource list. There must be at least one <ncwDataSourceEntry>.
  ! - name: The datasource name. Must correspond to the name specified in
  !   <ncwDataSourceDefinition>.
  !-->
<ncwDefaultDataSourceList>
  <ncwDataSourceEntry name="<Virtual failover aggregation pair name>"/>
</ncwDefaultDataSourceList>

<!--
  ! Configuration for a datasource with/without failover.
  !-->
  <ncwDataSourceDefinition type="singleServerOSDataSource" name="<Virtual Ag-
gregation pair name>" enabled="true">
```

where <Virtual Aggregation pair name> is the name of the aggregation pair as defined in the \$NCHOME/etc/omni.dat settings on both primary and backup object servers

and edit the following lines:

```
<ncwPrimaryServer>
  <ncwOSConnection host="<Primary Object Server host>" port="4100" ssl="false"/>
</ncwPrimaryServer>
<ncwBackUpServer>
  <ncwOSConnection host="<Backup Object Server host>" port="4100" ssl="false"/>
</ncwBackUpServer>
```

Remove comment lines before and after the backup object server definition.

The following lines define the ID and password on the Object Server for WebGUI to communicate.

Note: Ensure that 'smadmin' ID is created on the Object Server, with "System" privileges. This is an ID on the Omnibus Object Server application and not the Operating System ID.

```
<ncwDataSourceCredentials
```

```

        userName="smadmin" password="<pswd_OS>"
        encrypted="false"
    />

```

where <pswd_OS> is the password for smadmin into ObjectServers (Primary and Backup).

This file need not be altered unless there is a requirement to add or change the data sources.

Note that Web GUI server can communicate to more than one object server though the events from different object servers are not listed in a single WebGUI Event Viewer.

Stop and restart WebGUI.

6.2.1 **Secure Communication between WebGUI and aggregation pair**

6.2.1.1 *SSL settings for WebGUI*

GSMA deployments will use “SSL with FIPS” method for secure communications. In order for Web GUI server to communicate to the Object Server in SSL mode (with FIPS), the object server Certificate Authority (CA) file has to be imported into the Web GUI system. This step is applicable only if Object Server is used as the Certificate Authority.

The steps below describe the procedure for importing Object Server CA & enabling SSL (with FIPS) on

Ensure that SSL is enabled on the Omnibus Server and Certificate Authority (CA) file (GSMA_<objectservershorthostname>_CA.arm) is created on the Object Server (Refer to chapter)

Download CA certificate file to the following directory on the Web GUI server:

JazzSM_WAS_Profile/profile/config/cells/JazzSMNode01Cell/nodes/

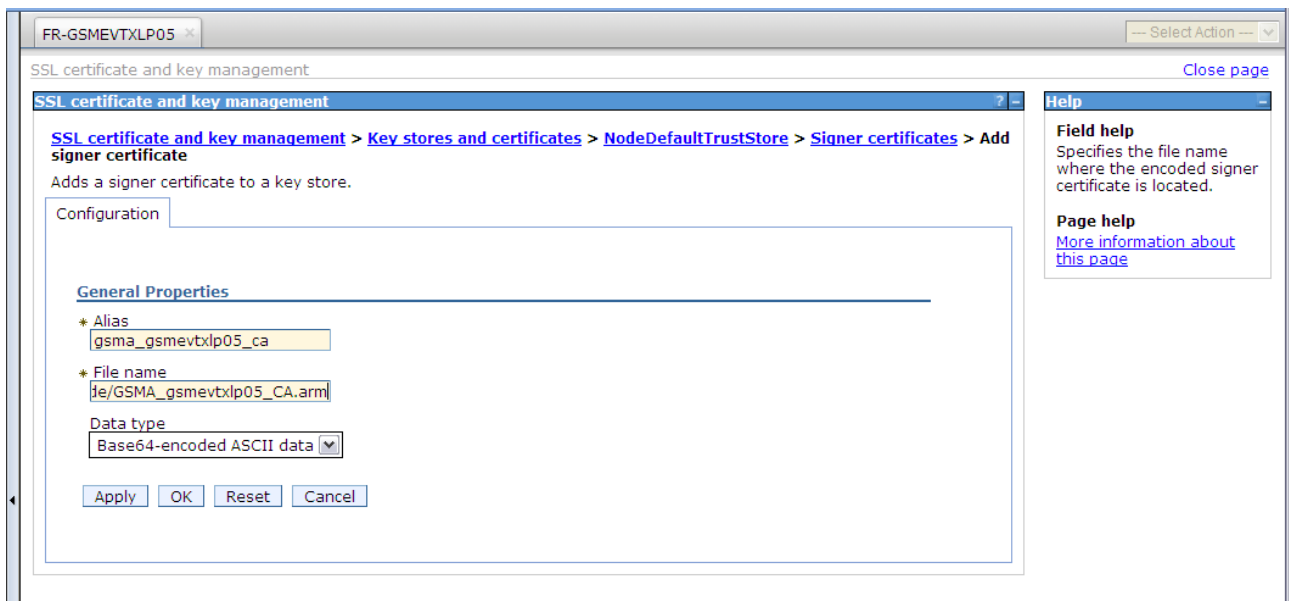
Add the signer certificate to the keystore so that it is recognized as a valid certificate.

Login as smadmin in WAS console,

1. Launch the administrative console and click Security > SSL certificate and key management.
2. On the "SSL certificate and key management" page, click Key stores and certificates. On the page that appears, click NodeDefaultTrustStore in the table at the center of the page.
3. Click Signer Certificates. On the page that appears, click Add.
4. Complete the fields in the "Configuration" panel as follows:



Enter the following values:



Alias Name : gsma_<objectservershorthostname>_ca (Example: gsma_gsmevtxlp05_ca)

File Name : The name and location of the CA certificate file from the Object Server

Example:

/opt/IBM/JazzSM/profile/config/cells/JazzSMNode01Cell/nodes/GSMA_gsmevtxlp05_CA.arm

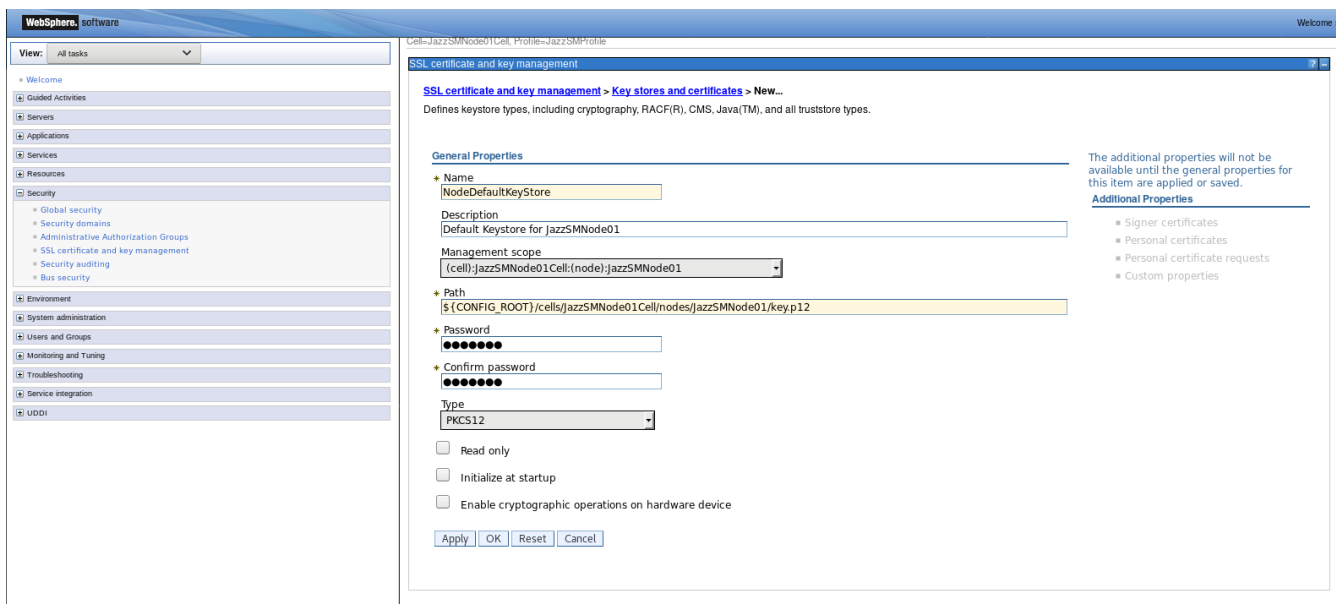
Data Type : Base 64-encoded ASCII data

5. Click Apply.
6. On the "SSL certificate and key management" page, click Save.

Update the NodeDefaultTrustStore password:

Make sure that you don't use the default password (WebAS) for the keystores/truststores.

Login as smadmin in WAS console, launch to WebSphere Administrative Console. In administrative console open Security > SSL certificate and key management > Key stores and certificates. Change default passwords for NodeDefaultKeyStore and NodeDefaultTrustStore.



Change the password and press "OK".

- 1) Backup the server init file /opt/IBM/netcool/omnibus_webgui/etc/server.init to a different name (server.init-datestamp) under the same directory.

- 2) Update server.init file with the following entries:

```
webtop.ssl.trustStorePassword:<NodeDefaultTrustStorePSWD>
```

where <NodeDefaultTrustStorePSWD> is the password that has been set for NodeDefaultTrustStore in previous step.

- 3) Backup the configuration file to a different name (ncwDataSourceDefinitions.xml-datestamp) under the same directory.

```
/opt/IBM/netcool/omnibus_webgui/etc/datasources/ncwDataSourceDefinitions.xml
```

- 4) Define Object Server port to be used for SSL communication. Change the port number and ssl values, in the section as shown below (highlighted)

```
<ncwPrimaryServer>
  <ncwOSConnection host="<Primary Object Server host>" port="4500" ssl="true"/>
</ncwPrimaryServer>
<ncwBackUpServer>
  <ncwOSConnection host="<Backup Object Server host>" port="4500" ssl="true"/>
</ncwBackUpServer>
```

- 5) Restart WebGUI Server.
- 6) Test SSL connectivity by performing a logon to the Web GUI server and access all functions from the Web GUI.

6.2.1.2 *Encrypting passwords using FIPS 140-2 mode encryption (optional)*

Generate AES_FIPS Encrypted passwords for default trust store password, defined in server.init file. Current default trust store password, which is not changed after installation, is stored in the server.init file:

The default password defined by Websphere application is "WebAS". If the password has been changed using the TIP GUI interface, please use the new password.

- 1) Use the following command to generate encrypted password for above value :

```
$NCHOME/omnibus_webgui/bin/ncw_fips_crypt -password <pswd_defaulttruststore> -key
$NCHOME/omnibus_webgui/etc/encrypt/vault.key
```

where <pswd_defaulttruststore> is the password that has been set previously for NodeDefaultTrustStore.

- 2) Backup the server init file /opt/IBM/netcool/omnibus_webgui/etc/server.init to a different name (server.init-datestamp) under the same directory.

- 3) Update server.init file with the following entries:

```
webtop.password.encryption:fips
webtop.fips:on
fips.security.key=%%/etc/encrypt/vault.key
webtop.ssl.trustStorePassword:<generated_pswd>
webtop.ssl.trustManagerType:IbmX509
webtop.ssl.trustStoreType:PKCS12
```

where <generated_pswd> is the WebSphere Trust Store password encrypted in step 1.

Then, follow the steps below to generate AES_FIPS encrypted passwords on the Web GUI server:

Generate AES_FIPS encrypted passwords for Object Server authentication, defined in ncwData-SourceDefinitions.xml file.

- 4) Ensure that \$OMNIHOME/etc/<objectservername>.props file on the Object Server is updated with the following entry:

ConfigCryptoAlg: 'AES_FIPS'

Document: GSMA_OMNibus_Failover_Cookbook.odt
Owner: Florence Rouanet-Rose

Date: 2019/10/30
Status: FINAL

Subject: Failover NOS Installation Cookbook

Page 51 of 82

- 5) Use the following command to generate encrypted passwords :

```
$NCHOME/omnibus_webgui/bin/ncw_fips_crypt -password <pswd_OS> -key  
$NCHOME/omnibus_webgui/etc/encrypt/vault.key
```

Where <pswd_OS> is the password of smadmin in objectserver.

- 6) Copy/Type the resulting password and update ncwDataSourceDefinitions.xml file as shown below:

```
<ncwDataSourceCredentials  
    userName="smadmin" password="<generated_psw>" encrypted="true"  
    algorithm="FIPS"  
>
```

where <generated_psw> is the password value generated using the ncw_fips_crypt utility in step 5.

- 7) Restart WebGUI Server.
- 8) Test SSL connectivity by performing a login to the Web GUI server and verify connectivity to Object Server by initiating Event Viewer list.

6.3 Configure Impact to connect to the virtual aggregation pair

- 1) Select the Data Model→ defaultObjectServer data type and edit settings.
- 2) Enter the primary and backup object servers following next definitions:

IBM Tivoli Netcool/Impact 7.1.0.9

Welcome | Data Model | Policies | Services | Operator View | Event Isolation and Correlation | Maintenance Window | Reports

AlertStatus x defaultobjectserver x

General Settings:

Provide general information which describes the data source. An * indicates required fields.

* Data Source Name: defaultobjectserver

* Username: ncosys

Password:

Maximum SQL Connection: 30

Database Failure Policy:

Select what action to take if Impact cannot connect to the database.

☒ Fail over

☐ Fail back

☐ Disable Backup

Primary Source:

Provide information on the primary database. * marks a required field.

* Host Name: gsmevtxlp05.lagaude.ibm.com

* Port: 4100

☐ SSL Mode

Test Connection

Backup Source:

Provide information on the backup database.

* Host Name: gsmevtxlp05.lagaude.ibm.com

* Port: 4100

☐ SSL Mode

Test Connection

Figure 4 - Impact setup on an OMNibus failover pair

Select “fail over” as Database Failure Policy. Enter hostnames and ports of each of the Object Server pair.

- 3) Setup the EIF Event adapter configuration file to point to EIF probes. In general, there should be probes running on the Primary and Backup ObjectServer that should be used for sending local events created by the GSMA solution.

The config file should be named `local_eif.conf` and be located in the `/opt/IBM/GSMA/config` directory. You can make a copy of `local_eif.conf.sample` as the starting point. Change the `ServerLocation` to the IP address of the Primary ObjectServer then the address of the Backup ObjectServer separated by comma. Change the Port by adding the port of the Backup Object Server.

Example (`local_eif.conf`):

```
# EIF Adapter Conf to send event to local EIF probe
ServerLocation=gsmevtxlp05.lagaude.ibm.com,gsmevtxlp04.lagaude.ibm.com
ServerPort=5529,5529
EventMaxSize=4096
```

BufferEvents=YES

BufEvtPath=/opt/IBM/GSMA/cache/local_eif.cache

6.3.1 Using SSL Connection to the ObjectServer

The Impact server to ObjectServer connection can also be setup to use SSL. When the Impact server and ObjectServer are in the same gSNI segment, non-ssl connections can be used. Otherwise SSL connections must be used per the ESA certification.

By default Impact connections are done using the port 4100. When setting up an ObjectServer SSL connection, the port 4500 must be used instead.

To establish SSL connection between the Impact server and the ObjectServer, the following steps must be applied.

It is to be noticed that the ObjectServer is seen from Impact as an "objectserver" Data Source Adapter (DSA) among other DSAs.

The below flows is specific to the ObjectServer DSA, if other DSAs need to be deployed, different flows may apply (and will be documented at the same time of the new DSA installation guide, if needed).

6.3.1.1 Importing the ObjectServer certificate

As for other Netcool Clients, Complete the following steps on the Impact Server.

a. FTP or Copy the root digital certificate (.arm file) from the ObjectServer server to the Impact Server and import to the Impact Server truststore.

1. To enable SSL connections between Netcool/Impact servers and external servers, you need to obtain a signed certificate for the external server, copy it to the Netcool/Impact server host machine and import it into Netcool/Impact's truststore:

```
<$IMPACT_HOME>/sdk/bin/keytool -importcert -alias unique_string -file
path_to_certificate_file -keystore
<$IMPACT_HOME>/wlp/usr/servers/<instance>/resources/security/trust.jks
-storepass impactadmin_password
```

Example :

```
<$IMPACT_HOME>/sdk/bin/keytool -importcert -alias gsma_gsmevtxlpv05_ca -file
/home/ncosys/LIU01.arm -keystore
<$IMPACT_HOME>/wlp/usr/servers/GSMA/resources/security/trust.jks -storepass
Tivoli4u
```

2. To enable SSL connections between Netcool/Impact GUI servers and external servers, you need to obtain a signed certificate for the external server, copy it to the Netcool/Impact GUI server host machine and import it into the Netcool/Impact GUI server truststore:

```
<$IMPACT_HOME>/sdk/bin/keytool -importcert -alias unique_string -file path
to_certificate_file -keystore
<$IMPACT_HOME>/wlp/usr/servers/ImpactUI/resources/security/trust.jks
-storepass impactadmin_password
```

Example :

```
<$IMPACT_HOME>/keytool -importcert -alias gsma_gsmevtxlpv05_ca -file  
/home/ncosys/LIU01.arm -keystore  
<$IMPACT_HOME>/wlp/usr/servers/ImpactUI/resources/security/trust.jks  
-storepass Tivoli4u
```

b. Restart the Impact Server.

Fill in the Alias Name you will use internally (we recommend to use the same name as the digital certificate name). The File Name is the file and path of the root digital certificate (.arm file) file you copied previously on your system. Select the Data-type based on the format used by the CA to create the certificate file and press the Apply button.

6.3.1.2 *Change the port used to communicate with ObjectServer pairs:*

In the ObjectServer data source, select the SSL Mode check box and check that you are using the appropriate SSL port.

IBM Tivoli Netcool/Impact 7.1.0.9

Welcome | **Data Model** | Policies | Services | Operator View | Event Isolation and Correlation | Maintenance Window | Reports

AlertStatus x defaultobjectserver x

General Settings:

Provide general information which describes the data source. An * indicates required fields.

* Data Source Name: defaultobjectserver

* Username: ncosys

Password:

Maximum SQL Connection: 30

Database Failure Policy:

Select what action to take if Impact cannot connect to the database.

☒ Fail over

☐ Fail back

☐ Disable Backup

Primary Source:

Provide information on the primary database. * marks a required field.

* Host Name: gsmevtxlp05.lagaude.ibm.com

* Port: 4500

☒ SSL Mode

Test Connection

Backup Source:

Provide information on the backup database.

* Host Name: gsmevtxlp05.lagaude.ibm.com

* Port: 4500

☒ SSL Mode

Test Connection

For each ObjectServer, change the port to “4500” and select “SSL Mode”.

6.4 Other Netcool Components

To configure controlled failback for clients that connect to the failover pair, perform the following steps for the client types:

6.4.1 Unidirectional ObjectServer Gateways

If unidirectional gateways are configured to connect to the virtual aggregation pair, disable fallback in the unidirectional collection layer gateways. In each collection ObjectServer Gateway properties file:

- Set Gate.Writer.Server to aggregation Object Server pair name (GSMAENV in example)
- Set Gate.Writer.FailbackEnabled to FALSE.

If unidirectional gateways are configured to connect the virtual aggregation pair to the display ObjectServers, disable fallback in the unidirectional display layer gateways. In each display ObjectServer Gateway properties file:

- Set Gate.Reader.Server to aggregation Object Server pair name (GSMAENV in example)
- Set Gate.Reader.FailbackEnabled to FALSE.

6.4.2 Bidirectional ObjectServer Gateways

Although not part of the proposed multitiered configuration, if bidirectional ObjectServer Gateways are used between the collection layer and aggregation layer, disable fallback in the bidirectional ObjectServer Gateways by using the following properties:

- Set Gate.ObjectServerB.Server to aggregation Object Server pair name (GSMAENV in example)
- Set Gate.ObjectServerB.FailbackEnabled to FALSE.

6.4.3 Event lists

If event lists are connecting to the virtual aggregation pair, disable fallback for event lists by setting the `-failbackpolltime` command-line option to 0 when running `nco_event` on UNIX® and Linux®, or `NCOEvent.exe` on Windows®.

If event lists are configured to connect to the display layer ObjectServers, and the event lists make a dual-write connection to the aggregation failover pair, start the event lists with the `-failbackpolltime` command-line option set to 0 so that they exhibit controlled fallback behaviour.

6.5 Configuring proxy servers for failover

6.5.1 Principle

The following chapter has been extracted from the Tivoli documentation and adapted to the configuration we put in example.

The proxy server failover setup requires the Tivoli Netcool/OMNibus basic failover architecture, and the following additional components: a primary proxy server and a backup proxy server.

The following figure shows the configuration for proxy server failover.

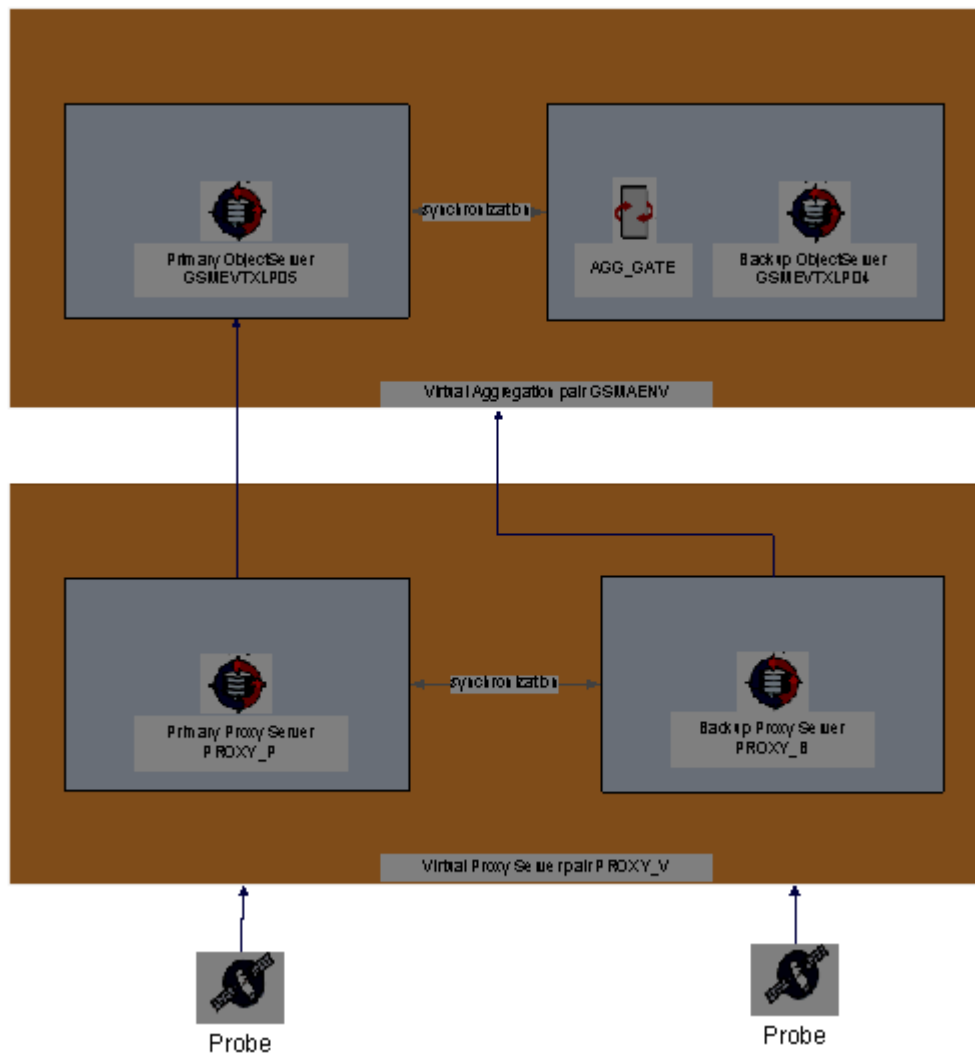


Figure 5 - Failover at probe/proxy level

In the basic failover setup, alert data from the primary aggregation ObjectServer is replicated in the backup aggregation ObjectServer through a bidirectional ObjectServer Gateway. If a connection to the primary aggregation ObjectServer fails, the clients attempt to connect to the backup aggregation ObjectServer. As shown in the figure, you must set up a virtual proxy server pair to which probes can connect. Set up the primary proxy server PROXY_P to have a single connection to the primary aggregation ObjectServer GSMEVTXLP05. Set up the backup proxy server PROXY_B for failover by configuring PROXY_B to connect to the virtual Object-Server pair GSMAENV.

Using the architecture shown in the preceding figure, configure the proxy servers for failover as follows:

- In the connections data file (\$NCHOME/etc/omni.dat or %NCHOME%\ini\sql.ini), set up the server communications details, using the following sample configuration as a guideline:

```
[GSMEVTXLP05]
{
    Primary: gsmevtxlp05.lagaude.ibm.com 4100 ssl 4500
```

```

}
[GSMEVTXLP04]
{
    Primary: gsmevtxlp04.lagaude.ibm.com 4101 ssl 4501
}
[GSMAENV]
{
    Primary: gsmevtxlp05.lagaude.ibm.com 4100 ssl 4500
    Backup: gsmevtxlp04.lagaude.ibm.com 4100 ssl 4500
}
[AGG_GATE]
{
    Primary: gsmevtxlp04.lagaude.ibm.com 4900
}

[PROXY_P]
{
    Primary:      nchost1 10003
}
[PROXY_B]
{
    Primary:      nchost2 10004
}
[PROXY_V]
{
    Primary:      nchost1 10003
    Backup:       nchost2 10004
}

```

- Configure probes to connect to the proxy servers. In the probe properties file:
 - o Set Server to PROXY_V.
 - o Set ServerBackup to "".

Results

With this configuration, if GSMEVTXLP05 fails, PROXY_P also fails, but the probes are automatically connected to PROXY_B, which will in turn connect to GSMEVTXLP04. If only PROXY_P fails, the probes will automatically connect to PROXY_B, and events will be sent to GSMEVTXLP05, which is still up and running as the primary ObjectServer.

6.5.2 Proxy server configuration

Configure the proxy servers by using the following instructions:

- 1) Create a dedicated user for the proxy in Omnibus Object Sever (ex.: proxy_<customer> - proxy_inf = Proxy User for Infrastructure). This user must be a member of group 'Probe'.

- 2) Logon to the Proxy server as ncosys
- 3) If the encryption key has not been already set on this server:
 - a) Login to server that will be running the Proxy process using the ncosys ID (This is required on the Object Server and any separate server running a probe or gateway).
 - b) Generate a random key to be used for encryption and store it in a file that only the userid that runs the OMNibus processes can access. GSMA recommends using file `$NCHOME/etc/security/keys/encryption.key`. Run the following command:


```
$OMNIHOME/bin/nco_keygen -o $NCHOME/etc/security/keys/encryption.key
```
 - c) Secure the file using:


```
chmod 600 $NCHOME/etc/security/keys/encryption.key.
```
- 4) To encrypt the password value for the proxy userid using the above key, use the `nco_aes_crypt` command from the ncosys userid.


```
$NCHOME/omnibus/bin/nco_aes_crypt -c AES_FIPS -k $NCHOME/etc/security/keys/encryption.key <password>
```

Where `<password>` is the password that you want to encrypt.

Example: @44:7h7OHG/NjaD4UfGgNEJOhMbL8ckp2ifiZBINfhgQSzo=@

- 5) Configure each proxy server following this example:
 - a) In the primary proxy server `PROXY_P` properties file, set `RemoteServer` to the primary ObjectServer name (in the example `GSMEVTXLP05`)


```
Name: '<Primary Proxy Name>'
AuthUserName: '<proxy user>'
AuthPassword: '<encrypted proxy user password>'
ConfigCryptoAlg: 'AES_FIPS'
ConfigKeyFile: '/opt/IBM/tivoli/netcool/etc/security/keys/encryption.key'
Connections: 100
MaxLogFileSize: 10240
MessageLevel: 'warn'
MessageLog: '/opt/IBM/GSMA/logs/ObjectServer/<Primary Proxy Name>.log'
PAServerName: 'NCO_PA'
RemoteServer: '<Primary ObjectServer>'
SecureMode: TRUE
```

where `<Primary Proxy Name>` is the name of the primary Proxy. In the example `PROXY_P`

`<proxy user>` is the user that has been created in ObjectServer for proxy in step (1)

`<encrypted proxy user password>` is the encrypted password of the proxy user encrypted by `nco_crypt` command.

`<Primary ObjectServer>` is the primary ObjectServer name. In the example the primary ObjectServer is `GSMEVTXLP05`.
 - b) In the backup proxy server `PROXY_B` properties file, set `RemoteServer` to the virtual aggregation pair object server (in the example `GSMAENV`)


```
Name: '<Backup Proxy Name>'
AuthUserName: '<Proxy user>'
```

```
AuthPassword: '<encrypted proxy user password>'
ConfigCryptoAlg: 'AES_FIPS'
ConfigKeyFile: '/opt/IBM/tivoli/netcool/etc/security/keys/encryption.key'
Connections: 100
MaxLogFileSize: 10240
MessageLog: '/opt/IBM/GSMA/omnibus/logs/ObjectServer/<Backup Proxy
Name>.log'
RemoteServer: '<Virtual Aggregation pair ObjectServer>'
SecureMode: TRUE
```

where *<Backup Proxy Name>* is the name of the Backup Proxy. In the example PROXY_B

<proxy user> is the user that has been created in ObjectServer for proxy in step (1)

<encrypted proxy user password> is the encrypted password of the proxy user encrypted by nco_crypt command.

Refer to the omni.dat definition for the *<Virtual Failover gateway pair name>*. In the example, the virtual gateway pair name is GSAMENV.

Configure Process Agent for Proxy start up

The Process Agent is required in the GSMA solution. It is used to start the Proxy Servers.

With the GSMA configuration, proxy servers will be restarted automatically if it goes down.

For each proxy server, add the startup of the proxy in Process Agent:

1) Add new process

```
nco_process 'ProxyServer'
{
    Command '$OMNIHOME/bin/nco_proxyserv -propsfile $OMNIHOME/etc/<Proxy Server
Name>.props -pa NCO_PA' run as 'ncosys'
    Host = '<Proxy Server Hostname>'
    Managed = True
    RestartMsg = '${NAME} running as ${EUID} has been restored on $
{HOST}.'
    AlertMsg = '${NAME} running as ${EUID} has died on ${HOST}.'
    RetryCount = 0
    ProcessType = PaPA_AWARE
}
```

2) Define a service:

```
nco_service 'Proxy'
{
    ServiceType = Master
    ServiceStart = Auto
}
```



```
process 'ProxyServer' NONE  
}
```

7. EIF probe Failover

GSMA recommend installing EIF probe on the TEMS server. If the TEMS is configured in High Availability, it is recommended to include the probe in this mechanism.

In case of probe failure, GSMA recommends installing a backup probe that will be used in case of failure of the primary EIF probe. That can be the local ObjectServer Probe used for Omnibus automation (triggers) and Impact to request updates on the ObjectServer tables (alert.status, journal, ..)..

It is recommended that the local ObjectServer probe be configured using the virtual aggregation pair, as any other probe.

7.1 Overview

The GSMA recommended as failover process to use the standard EIF communication failover. It allows using a running EIF Probe as Backup for load balancing some GSMA actions.

The EIF probe routing ITM6 events to Omnibus server is located on the TEMS server and the local ObjectServer Probe is running on the Primary ObjectServer for local event creation as recommended by GSMA architecture.

Both primary and backup probes have to be defined in a single Event destination on TEMS. EIF configuration allows defining a pair of list of servers as target location and list of port as location's port. In this case, the Event destination in TEMS is configured to route event to 2 different hosts (eventually two different ports).

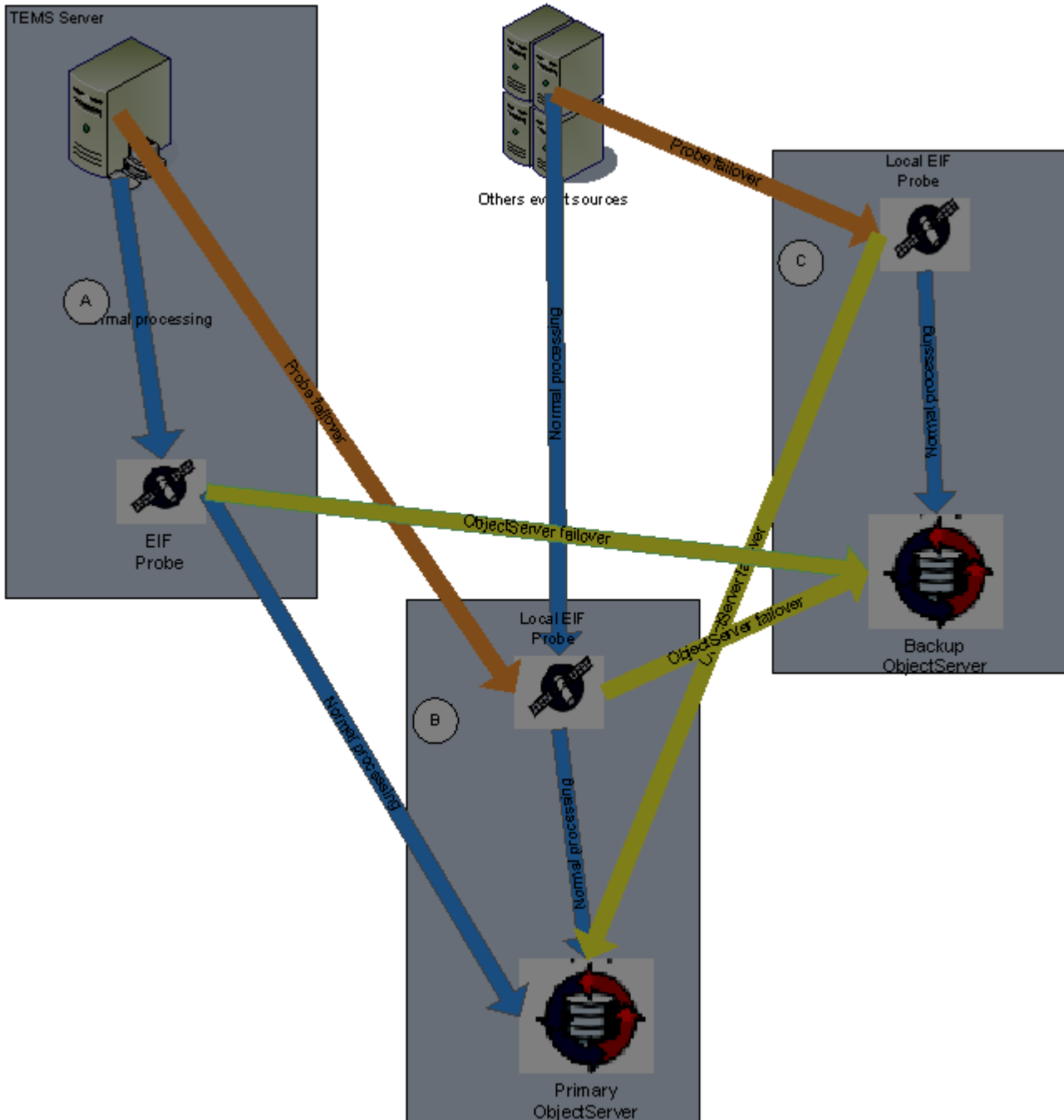


Figure 6 - EIF Probe failover - Flow

Both EIF and local (ObjectServers') EIF probes are defined through a same event destination label GSMAENV at TEMS layer. When primary probe (EIF Probe) is up and running, all ITM6 events are routed to this primary Probe (A in previous figure). When connection to primary EIF probe fails (A), automatically events are routed to Backup Probe layer (B).

When primary EIF Probe (A) is up and running, events are routed back to this EIF Probe.

Some events can be directly routed by other processing to the local EIF probe without impacting the failover process (they use the EIF postzmsg command). The local EIF probe on Backup ObjectServer must backup the local EIF probe on primary ObjectServer. All local_eif.conf file defined in GSMA components (Impact Server, Omnibus Server) must be updated to route events to Primary then to Backup local EIF probe.

Example (local_eif.conf):

```
# EIF Adapter Conf to send event to local EIF probe
ServerLocation=gsmevtxlp05.lagaude.ibm.com,gsmevtxlp04.lagaude.ibm.com
ServerPort=5529,5529
EventMaxSize=4096
BufferEvents=YES
BufEvtPath=/opt/IBM/GSMA/cache/local_eif.cache
```

For Backup ObjectServer local probe(C), Primary ObjectServer must also be defined as primary destination as for other component using postzmsg to send event to ObjectServer.

7.2 Configuring backup EIF probe

The backup EIF probe must be installed following the GSMA Omnibus cookbook installation recommendation and GSMA packages.

Please Refer to Omnibus Installation cookbook Section 7.

Configure EIF Probe to route events to the ObjectServers aggregation pair (see previous chapter).

Test Probe connection to the Omnibus aggregation pair.

```
/opt/IBM/GSMA/bin/postzmsg -f /opt/IBM/GSMA/config/local_eif.conf -m 'Test
Message from probe' -r CRITICAL TEC_Notice TEC
```

Check the event is received on Primary ObjectServer.

7.2.1 Setting up SSL & FIPS Certificate Database

If the session to the ObjectServer will be using SSL, you need to setup an SSL keystore database and import the root certificate for the Certificate Authority that issues the certificate for the Object Server. Refer to section Configure the Object Server in secure SSL mode for information on when SSL is required.

NOTE : If Probe which will be configured is placed on Primary Objectserver or Backup Objectserver , please omit these chapter if steps from chapter 5 “Secure SSL & FIPS Communication setting” were conducted.

The GSMA standard would be for the Object Server to have a self-signed certificate so you would need to import the CA certificate from the Object Sever.

To setup the database and import the CA certificate:

- 1) Login to ncosys ID.
- 2) Create the certificate key database file using ikeyman. The command line version is \$NCHOME/bin/nc_gskcmd. The key database must be called omni.kdb and located in the \$NCHOME/etc/security/keys directory. Create the key database by issuing:

```
$NCHOME/bin/nc_gskcmd -keydb -create -type cms -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw> -expire 7300 -stash
```

This will create the key database using the assigned password which will expire in 20 years. The assigned password can be anything and has no relation to the ncosys password or any object server ID/passwords.

Note: Ensure that the assigned password is stored in secure department database or other sources for future use.

Note: According to new IBM password security policy, please ensure the passwords length is set with longer than 15 (≥ 15) for all user IDs. When you generate the 15 character's password by manual or password management tool, please exclude some specific characters (e.g., '"\>^\$&([,],{,},<,>,(,))

- 3) If a commercial CA is being used, the CA certificate may already be in the new certificate database. If not, you will need to import it similar to how the self-signed certificate is imported below.

Obtain the root CA certificate from the Object Server and place it in the \$NCHOME/etc/security/keys directory. The GSMA standard name for a self-signed certificate is GSMA_<Object Server Short Hostname>_CA.arm

Import the CA certificate using command:

```
$NCHOME/bin/nc_gskcmd -cert -add -db $NCHOME/etc/security/keys/omni.kdb -pw
<your_assigned_pw> -label GSMA_<Object Server Short Hostname>_CA -file GSMA_<Object Server
Short Hostname>_CA.arm -fips
```

Where:

<**your_assigned_pw**> is the password you assigned to the omni.db key DB.
 <**Primary Object Server Short Hostname**> is the short hostname of the Object Server and **GSMA_<Object Server Short Hostname>_CA** is the name of the CA certificate get from Primary ObjectServer which is signer.

Example :

- 4) \$NCHOME/bin/nc_gskcmd -cert -add -db "\$NCHOME/etc/security/keys/omni.kdb" -pw tivo-liNetCool@zxc -label GSMA_gsmevtxlp08_CA -file /opt/IBM/tivoli/netcool/etc/security/keys/GSMA_gsmevtxlp08_CA.arm -fips

You can list all the CA certificates in the db using:

```
$NCHOME/bin/nc_gskcmd -cert -list all -db $NCHOME/etc/security/keys/omni.kdb -pw <your_assigned_pw>
```

7.3 Configuring new Event destination on TEMS

Logon TEMS server using 'root' or the recommended logon used for TEMS management.

Logon HUB TEMS using :

```
tacmd login -s <TEMS hostname> -u sysadmin -p <sysadmin password>
```

To list all event destinations use command:

```
tacmd listeventdest
```

Create a new event destination on HUB TEMS using the following command:

```
tacmd createEventDest -i 2 -p host1=<primary probe hostname>:5529 host2=<secondary probe hostname>:5529 name=GSMAENV
```

Example:

```
tacmd createEventDest -i 2 -p host1=tsalpar3.lagaude.ibm.com:5529
host2=gsmevtxlp05.lagaude.ibm.com:5529 name=GSMAENV
```

KUICCE004I: Are you sure you want to create the event destination server definition GSMAENV with server ID 2 on the server?

Enter Y for yes or N for no: Y

KUICCE007I: The event destination server definition GSMAENV with server ID 2 was successfully created on the server at <https://tsalpar3.lagaude.ibm.com:3661> .

To modify your event destination as default Event Destination:

```
tacmd editeventdest -i 2 -p NAME=GSMAENV  
host1=tsalpar3.lagaude.ibm.com:5529 host2=gsmevtxlp05.lagaude.ibm.com:5529  
DEFAULT=Y
```

Command to view a specific event destination:

```
tacmd vieweventdest -i <id>
```

```
[tsalpar3:root:/home/root:] tacmd vieweventdest -i 2
```

Server Id : 2

Server Name: GSMAENV

Server Type: TEC

Description:

Default : N

Host1 : tsalpar3.lagaude.ibm.com:5529

Host2 : gsmevtxlp05.lagaude.ibm.com:5529

Host3 : Not set

Host4 : Not set

Host5 : Not set

Host6 : Not set

Host7 : Not set

Host8 : Not set

After modification issue this command to update information without recycling TEMS:

```
tacmd refreshtecinfo -t eif
```

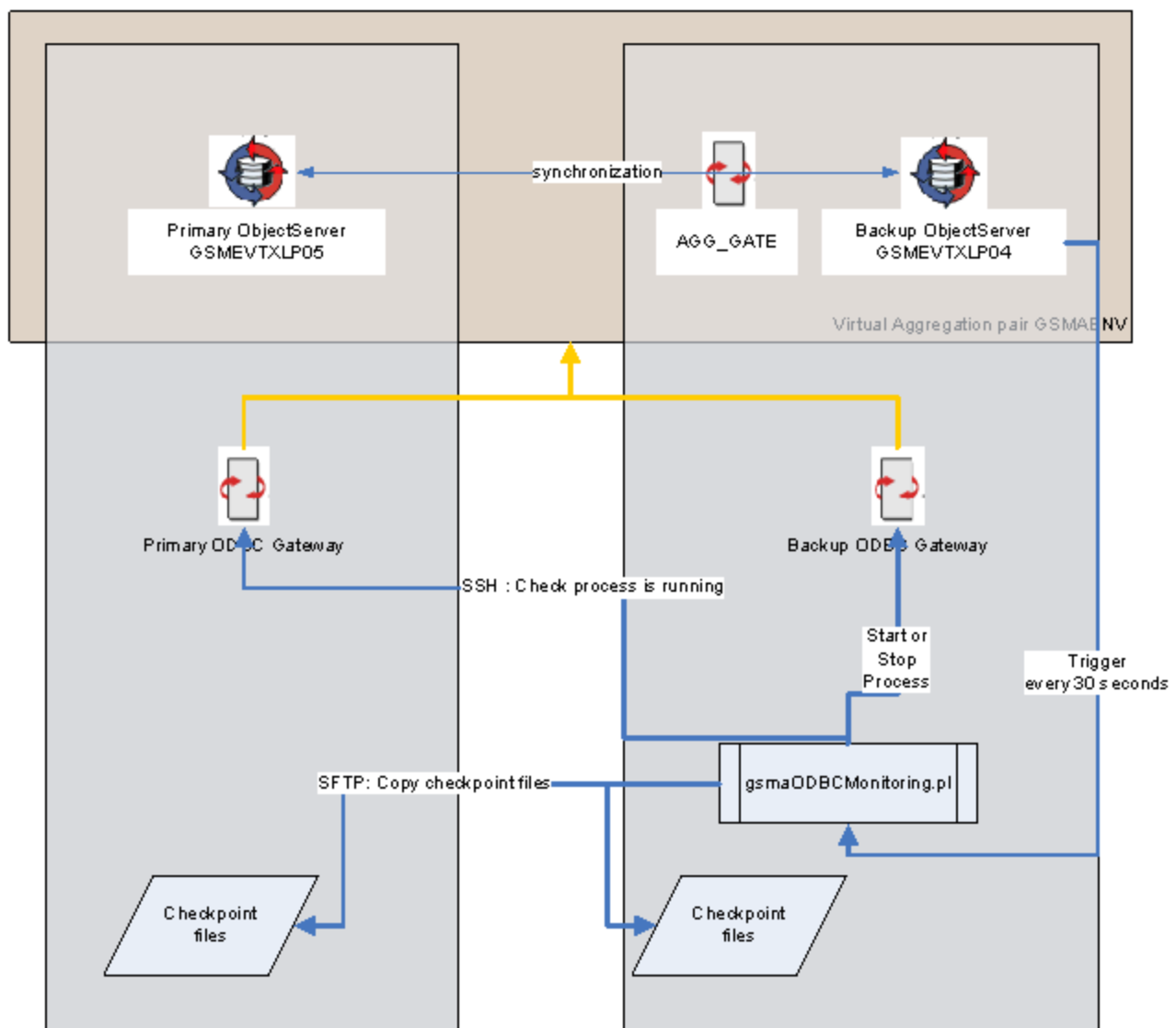
8.GSMA solution for TDW Gateway Failover

Tivoli does not provide any redundancy or failover solution for the Gateways. This development is not in Tivoli plans in short terms.

8.1 Overview of the GSMA solution

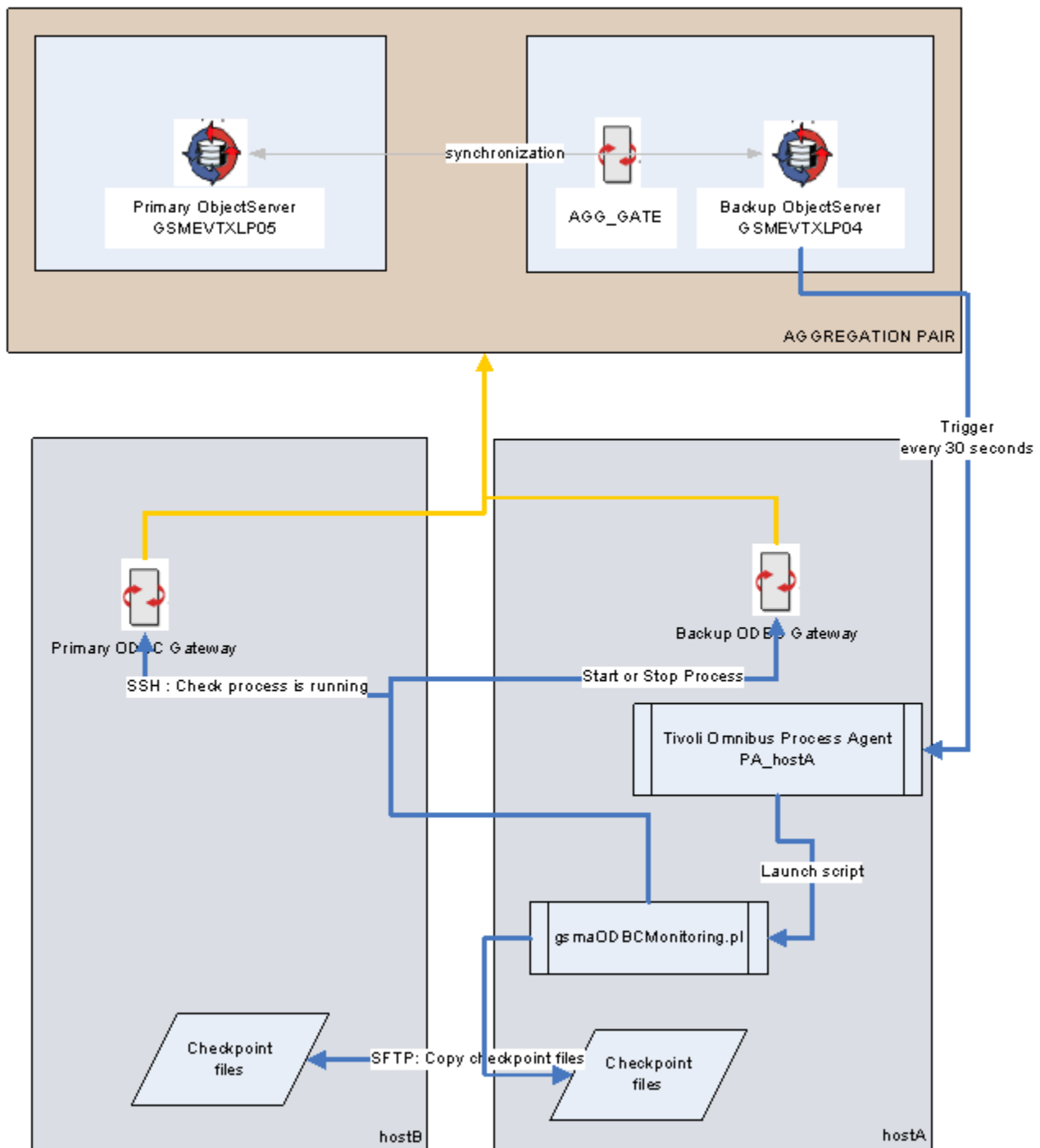
The solution must be implemented in an Omnibus Failover environment.

In this environment, two TDW gateways are installed on both Primary and Backup Object-Servers, this is the GSMA recommendation.



Both TDW gateways are configured to connect the virtual aggregation pair.

In some situations, these TDW gateways could also be installed on another redundant place such as the hosts used for TDW databases.



The TDW gateway installed on primary is called Primary TDW gateway. The other one is the Failover TDW gateway.

In normal situation, when Primary Object Server is running, Primary TDW gateway is running. Failover TDW gateway process is down.

When the Primary TDW gateway process is stopped then the Failover TDW gateway process is automatically started until the Primary is up again.

8.1.1 Detailed TDW Gateway Failover solution content

On Backup ObjectServer, an Omnibus temporal trigger “gsmaTDWGatewayFailover” every 30 seconds starts an external procedure “gsmaTDWGatewayMonitoring” that in turn triggers the monitoring script “gsmaTDWGatewayMonitoring.pl” on the Failover host (localhost if the gateway is installed on backup ObjectServer) using ncosys user (id 997, group 997). In case the failover gateway would be installed on another place than the backup ObjectServer, then the script must run on this system but it must be triggered from the backup ObjectServer. That means the gateway system’s process agent must be known from the backup ObjectServer in order to trigger the script using the process agent (omni.dat).

The monitoring script checks the Primary gateway process is running on Primary OS using \$OM-NIHOME\bin\ncs_ping utility.

- If the primary is down
 - o If the Failover is not running, the script get the gateway cache files from Primary to failover in order to avoid trying to include duplicate entries when failover gateway process will start next time.
 - o If the failover is not already running, then the Failover Gateway process is started by script. A warning event is sent to Omnibus in order to inform operator the gateway has switched.
- If status is up, the script checks the gateway is not running on Failover.
 - o If gateway is running on failover, the script stops it. It clears the warning event about Primary to Failover Gateway switch.
 - o If the Failover is not running, the script get the cache files from Primary to failover in order to avoid trying to include duplicate entries when failover gateway process will start next time.

8.1.2 SSH tunnel

The GSMA TDW gateway failover solution uses a SSH tunnel and SFTP to transfer gateway cache files from one this system to the other.

8.1.2.1 Public key authentication

Public key authentication is one of the most secure methods used to authenticate when using a Secure Shell. Public key authentication uses a pair of computer generated keys, one public and one private. The public key can be distributed and resides in the SFTP server. The private key is unique to the user and must not be shared.

An identity consists of two parts, called the private key and the public key². The private key represents your identity for outgoing SSH connections. When you run an SSH client, such as ssh or scp, it requests a connection with an SSH server, the client uses the private key to prove your identity to the server. The public key represents your identity for incoming connections to your account. When an SSH client requests access to your account, using a private key as proof

² The same key can be shared between the server and client; then it is told about symmetric keys; here asymmetric keys are used, safer but with an additional processing overhead affordable with a limited usage.

of identity, the SSH server examines the corresponding public key. If the keys match, authentication succeeds and the connection proceeds.

Passphrase is an optional property that is used to provide extra protection for the private key. A null passphrase may be used for system-to-system authentication, as long as the `authorized_keys` or `authorized_keys2` file limits access only from specific hosts by specifying the "from" option with the appropriate value. Access to Group 1 systems using null-passphrase keys may only be initiated from systems managed as Group 1 systems. In order to prevent the keys from being used for interactive user logins, the private key file on the originating hosts must be owned by application and system users/groups, which do not have remote, password-authenticated login capability, and may only be readable and writable by the file owner.

The key-pair can be generated using any third party service and you can choose any of the standard encryption algorithms. ITCS104 recommend using RSA.

In order to perform a secured file transfer OpenSSH must be installed on both servers (extractor: LDS server and receiver: repository server). OpenSSL is a prerequisite to OpenSSH. These products are available on the AIX Bonus Pack CD.

8.1.2.2 GSMA users and groups definitions

The objective of implementing Secure File Transfer and Secure Shell is to ensure a proper authentication of clients connected to the remote OS.

The GSMA user used to push or pull files or execute remote commands must be able to take these actions without having to provide a password.

8.1.3 Solution limitations

During a maximum time frame of 30 seconds, Primary and Failover TDW gateway processes could be running simultaneously. In this case, some same events could be reported from both TDW Gateways that could try to insert duplicate events. In this case, some error messages could be seen at DB2 and TDW Gateway log layer. But anyway, no duplicate events will be reported.

During a minimum time windows of 30 seconds, Primary and Failover TDW gateway processes could be down simultaneously but that is enough to avoid losing events.

There is no copy back of the gateway cache files from backup to primary when primary is up again. If the primary TDW gateway is stopped a long time, numbers of events will be sent back to the reporter DB that could have already been sent by failover.

8.2 TDW Gateways settings

On both Primary and Backup ObjectServers, an TDW Gateway must be installed following the GSMA TCR/TDW recommendations.

Please refer to the [GSMA Netcool TDW TCR Reporting](#) document.

1. Check the "Gate.RdrWtr.Server" parameter's value in `G_JDBC.props` gateway file is the ObjectServer pair name (GSMAENV in GSMA examples) on both TDW gateway settings.
2. For TDW failover gateway, do not include the automatic startup of the TDW gateway in `inittab` but create anyway the script to start this TDW gateway as described in the GSMA Netcool TDW TCR Reporting document.

8.3 Settings on system hosting TDW Primary Gateway

These settings have to be made on the PRIMARY system that hosts TDW Primary Gateway. By default, this is the Primary ObjectServer but in some configurations this TDW Gateway could be installed on any other third part system (TDW layer for example).

8.3.1 Users Specifications

A user ncosftp must be created because the standard GSMA administration user, ncosys, is not allowed for remote connection that is used to transfer checkpoint files from primary to failover gateway.

Groups	
ncosftpg (used on primary server)	<ul style="list-style-type: none"> Group ID = 1001 <i>ncosftpg is a specific group name that must be used unix command execution uniquely. (ps)</i>
ncoadmin	<ul style="list-style-type: none"> Group ID = 997 <i>ncoadmin is the default GSMA group should already has been created when the Omnibus environment has been installed.</i>
IDs	
ID ncosftp Used on Primary Server	<ul style="list-style-type: none"> User ID = 1001 Set Primary GROUP to "ncosftpg" Assign "ncoadmin" group to ncosftp "Group SET" Set "user can LOGIN" to false Set "user can LOGIN REMOTELY" to true (user must access user via sftp). umask shall be set at default for ncosftp user, which is 022 <i>This id will be used to execute unix ps command to check odbc gateway process is running.</i>
ID ncosys	<ul style="list-style-type: none"> User ID = 997 Set the password for userid ncosys to be non-expiring Set "allow remote login" to false. (use "su - ncosys" to login to this ID when installing products). Set group to "ncoadmin" (umask shall be set at default for ncosys user, which is 022) <i>This id will be used for Omnibus installation, upgrades, configuration and also for running the Object Server and Probe. It should already has been created when the Omnibus environment has been set.</i> <i>According to new IBM password security</i>

	<i>policy, please ensure the passwords length is set with longer than 15 (≥ 15) for all user IDs. When you generate the 15 character's password by manual or password management tool, please exclude some specific characters (e.g., ""\>^\$&([, {, }, <, >, (,))</i>
--	--

8.4 Settings on system hosting TDW Failover Gateway

These settings have to be made on the FAILOVER system that hosts TDW Failover Gateway. By default, this is the Backup ObjectServer but in some configurations this TDW Gateway could be installed on any other third part system.

8.4.1 User Specifications

The user ncosys (id 997) and group ncoadmin (id 997) must already have been created when the Omnibus environment has been installed.

8.4.2 SSH Tunnel settings to transfer files

The transfer of cache files from Primary to Failover system is based on secure FTP (SFTP) and SSH. It is initiated from the FAILOVER system (get mode).

By default, the requestor user is ncosys (on failover) and target user is ncosftp (on primary). Please note that ncosftp user is also used in PushToProbe solution.

For the commands listed below we make the following assumptions:

1. The system used to initiate the transfer is the system where is installed the TDW failover gateway
2. ncosys is the requestor user.
3. \$remoteuser is the target user that will access the cache files of TDW gateway, by default ncosftp.
4. \$clienthost is the Failover full qualified hostname (where is installed the failover gateway).
5. \$remotehost is the Primary full qualified hostname (where is installed the primary gateway).

8.4.2.1 Public/Private Key Generation on the client (Failover side)

Logon ncosys.

Generate a public key (if it has not being already created for PushtoProbe solution):

```
export clienthost=<Failover hostname>
export remoteuser="ncosftp"
export remotehost=<Primary hostname>
```

```
$ export clienthost=gsmevtxlp04.lagaude.ibm.com
$ export remoteuser="ncosftp"
$ export remotehost=gsmevtxlp05.lagaude.ibm.com
```

```
ssh-keygen -t rsa
chmod 700 .ssh
cd .ssh
chmod 600 id_rsa id_rsa.pub known_hosts
```

```
$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/ncosys/.ssh/id_rsa):
Created directory '/home/ncosys/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/ncosys/.ssh/id_rsa.
Your public key has been saved in /home/ncosys/.ssh/id_rsa.pub.
The key fingerprint is:
c1:bb:c3:80:93:a7:22:ed:1a:b1:9a:7f:35:7d:0e:53 ncosys@gsmevtxlp05
```

The command above has generated a public/private key pair.

Do NOT enter a passphrase. That's kind of important, because when a passphrase has been defined, each connection above will need this passphrase to be entered in order to use the private key. For an automated solution, the private key must not have a passphrase. This is important: by not placing a passphrase on the private key, the security implication is that mere *possession* of the private key is sufficient to gain access to what ever resources into which you've placed the corresponding public key. Safeguard your private key.

The **private** key was placed in /home/ncosftp/.ssh/id_rsa. This needs to be kept secure, because of the security implication above, but also needs to be available to the process attempting to make an ssh, sftp or rsync connection. If these tools are run under the 'ncosftp' account, the tools will automatically look in the ".ssh" directory and I won't need to specify the private key location. Otherwise, command line options will need to point to the right place and key.

The **public** key is in /home/ncosftp/.ssh/id_rsa.pub. This is the key that gets distributed to those places that want to grant ncosftp user access.

Limit access from the specific client:

```
perl -i -ple 's/^/from=""$clienthost" /' /home/ncosys/.ssh/id_rsa.pub
```

```
$ perl -i -ple 's/^/from=""$clienthost" /' /home/ncosys/.ssh/id_rsa.pub
$ cat /home/$remoteuser/.ssh/id_rsa.pub
```

```
from="tsadev3inttec.lagaude.ibm.com" ssh-rsa AAAAB3NzaC1yc2EAAAABIwAAAIEAvi-
IcGCz/3MMCeQL9bXg41cZrGCfHMqDYzFA4EtV/2lAbegBCP/l4yIsW5C8Hbu21d/GCf6jN-
gR5if6VsRH6xjZu0lkUhd8T2sJ3uapAVrDnmC1Ra1BbU0UxqLaZwnfjdXqTp2r0Jcoanh+J0YggyP-
Ih5BfW/fh0JGu1Mf1d0N+U= ncosftp@tsadev3inttec
```

8.4.2.2 Transferring the public key to the target repositories

On the "remote" repository server, using **ncosftp** in that account's home directory, create a ".ssh" subdirectory, and in that directory create a new text file called "authorized_keys". If it already exists, use the existing file.

The directory must be chmod 700, and the file 600. In other words, only the owner can access the directory, and the file within it.

Create .ssh directory if it does not exist on the **\$remotehost**:

```
ssh $remoteuser@$remotehost " [ ! -d /home/$remoteuser/.ssh ] && mkdir /home/
$remoteuser/.ssh"
```

```
$ ssh $remoteuser@$remotehost "[ ! -d /home/$remoteuser/.ssh ] && mkdir /home/
$remoteuser/.ssh"
```

The authenticity of host 'gdsmevtxlp04.lagaude.ibm.com (9.100.70.20)' can't be established.

RSA key fingerprint is d1:fe:64:fa:22:4a:31:c7:2d:fa:76:93:08:ed:50:fd.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'tsadev3contecb.lagaude.ibm.com,9.100.70.20' (RSA) to the list of known hosts.

```
ssh $remoteuser@$remotehost "chmod 700 /home/$remoteuser/.ssh"
```

Copy public key to repository server \$remoteuser@\$remotehost:

```
scp id_rsa.pub $remoteuser@$remotehost:id_rsa_tmp.pub
```

```
$ scp id_rsa.pub $remoteuser@$remotehost:id_rsa_tmp.pub
```

ncosftp@gdsmevtxlp04.lagaude.ibm.com's password:

```
id_rsa.pub
```

```
100% 268 0.3KB/s 00:00
```

Add the public key to the authorized_keys file:

```
ssh $remoteuser@$remotehost "cat id_rsa_tmp.pub >> /home/$remoteuser/.ssh/author-
ized_keys"
```

```
ssh $remoteuser@$remotehost "chmod 600 /home/$remoteuser/.ssh/authorized_keys"
```

```
ssh $remoteuser@$remotehost "rm id_rsa_tmp.pub"
```

Once saved anyone in possession of the private key that matches this public key can now login as this account.

8.4.2.3 Test SFTP Connection

The public key has been transferred in account ncosys on \$remotehost server. So now, on Primary OS server, logged in as \$remoteuser user, and where the private key is stored as described above, an sftp session looks like this:

```
sftp $remoteuser@$remotehost
```

\$remoteuser specifies the remote account on \$remotehost to login as.

The private key is matched to the public key, which indicates \$remoteuser is authorized to login to that account. The sftp session has been created. No interactivity required.

8.4.3 Install and Configure the GSMA TDW Gateway Monitoring script

Use the following steps to configure the GSMA TDW Gateway Monitoring script.

To configure the GSMA TDW Gateway Monitoring script:

1. Logon to the system hosting the TDW failover Gateway with **ncosys** ID.
2. Download the gsma_NOS_failover_x.x.x.rpm package (version 9.0 or above) from <http://gsma.lagaude.ibm.com/packages>
3. Place the gsma_NOS_failover_x.x.x.rpm file in the /tmp directory.
4. Install the rpm with command

```
rpm -dbpath /opt/IBM/GSMA/rpmdb -ivh /tmp/gsma_NOS_failover_<version>.rpm -ig-noreos --noscripts
```

5. You must add a primary TDW gateway server entry to the interfaces file. A name of this entry will be used by the Monitoring script to pass into nco_ping utility in order to check the state of the primary TDW gateway.
 - Edit the \$NCHOME/etc/omni.dat data file.
 - Add following entry to the the file

```
[G_JDBC_PRI]
{
    Primary: <primary TDW gateway host> 4700
}
```

Notes:

- a name of the entry should not match the real name of the Primary TDW Gateway, because it is needed only for the nco_ping utility on the failover side (G_JDBC_PRI is a GSMA default)
 - <primary TDW gateway host> is a name of the host where the primary gateway is installed
 - port number must match the port number of the primary gateway, which is set in the interfaces file on the primary host (4700 is a GSMA default)
 - Run \$NCHOME/bin/nco_igen to convert the interfaces file.
1. Create directory for TDW gateway cache files:

```
/opt/IBM/tivoli/netcool/omnibus/var/<TDW gateway name>
```

Default is /opt/IBM/tivoli/netcool/omnibus/var/G_JDBC.

2. Configure the Monitoring script by copying the `gsmaTDWGatewayMonitoring.conf.sample` to `gsmaTDWGatewayMonitoring.conf` and updating the configuration file `gsmaTDWGatewayMonitoring.conf` located in the `/opt/IBM/GSMA/failover/config` directory. Modifying the default parameters' values according to your configuration:
 - **PRIMARYHOST:** is the full qualified hostname of the system hosting the primary TDW Gateway as defined in SSH tunnel settings (see). There is no default value.
 - **PRIMARYUSERID:** userid used for SSH connection (see). Default `ncosftp`.
 - **GatewayName:** is a name of the TDW gateway. Default `G_JDBC`.
 - **PrimaryGatewayName:** is the name of the primary TDW gateway server entry in the interfaces file, which was created on the previous step. Default `G_JDBC_PRI`.
 - **GatewayProcess:** is a name of TDW Gateway executable. Default `nco_g_jdbc`.
 - **StartScript:** is the name of the shell script used to start the TDW gateway. Default `/opt/IBM/GSMA/bin/tdwgw.start`.
 - **DelayForSendingFiles:** is the delay in seconds, the checkpoint files are transferred from Primary to Failover server. The default value is 180 (every 3 minutes) but if the global size of the files exceeds 10MB, we recommend to increase this value to 600 (every 10 minutes) but we do not recommend to exceed one hour.
1. Check the Primary TDW Gateway is running and the Failover TDW Gateway is down.
2. Manually run the script in order to check the configuration:

`/opt/IBM/GSMA/failover/bin/gsmaTDWGatewayMonitoring.pl -v`

You should get the following result:

```
Checking TDW gateway G_JDBC_PRI is running
Rc for search 0 on command /opt/IBM/tivoli/netcool/omnibus/bin/nco_ping
G_JDBC_PRI
Process is running on primary
Checking TDW gateway G_JDBC is running
Rc for search 65280 on command /opt/IBM/tivoli/netcool/omnibus/bin/nco_ping
G_JDBC
Process is down on Failover side
Checking /opt/IBM/GSMA/failover/bin/gsmaTDWGatewayMonitoring.cnt
exiting /opt/IBM/GSMA/failover/bin/gsmaTDWGatewayMonitoring.pl
```

8.5 Backup ObjectServer settings

On backup side, the ObjectServer must have one additional trigger and one procedure installed and enabled.

To configure the backup ObjectServer:

1. Logon to the backup ObjectServer with **ncosys** ID.
2. Download the `gsma_NOS_failover_x.x.x.rpm` package (version 9.0 or above) from <http://gsma.lagaude.ibm.com/packages>
3. Place the `gsma_NOS_failover_x.x.x.rpm` file in the `/tmp` directory.
4. If it does not already exist, create failover directories


```
mkdir /opt/IBM/GSMA/failover
mkdir /opt/IBM/GSMA/logs/failover
```
5. Install the rpm with command

```
rpm -dbpath /opt/IBM/GSMA/rpmdb -ivh /tmp/gsma_NOS_failover_<version>.rpm -ig-  
noreos --noscripts
```

6. If the TDW failover gateway is not installed on the Backup ObjectServer itself,
 - Check the omni.dat file contains definition for the Process Agent installed on the system hosting the failover TDW Gateway.
 - Edit the /opt/IBM/GSMA/failover/install/gsma_TDW_failover_gateway.sql script and change the 'host' value by putting the hostname assigned to this Process Agent (refer to omni.dat).

1. Check the ObjectServer is running and include the GSMA SQL instructions file to be applied to this ObjectServer:

```
export objsrvr=<Failover Object Server Name>
```

```
$OMNIHOME/bin/nco_sql -user ncosys -server $objsrvr < /opt/IBM/GSMA/failover/in-  
stall/gsma_TDW_failover_gateway.sql
```

where Failover Object Server Name is the name of your Object Server (GSMEVTXLP04 in the example).

2. You can check the monitoring script is running successfully by setting the debug state to true for the gsmaTDWGatewayFailover trigger. In this case the gsmaTDWGateway-Monitoring.pl script will generate an output file in /opt/IBM/GSMA/logs/failover directory on the server running the script.

9. Upgrade GSMA code in failover mode

9.1 Upgrade ObjectServer Aggregation pair

In failover environment, only contents of Omnibus tables defined in the (AGG_GATE.tblrep.def) table replicate definition file for gateway are synchronized between Object Servers involved in the failover pair. For each of these tables, only fields listed in the gateway mapping file (AGG_GATE.map) are synchronized between Object Servers.

When a new table is created by GSMA or by new Omnibus ObjectServer version and this table must be synchronized between Primary and backup ObjectServers, this table must be added to the Gateway table replicate definition file.

GSMA tables used for immediate triggered action at event reception, gsmaChangeRequest or GSMATicketRequest for example, must not be synchronized between primary and backup ObjectServers.

When a new field is added in one of the ObjectServer table, the gateway mapping file must be upgraded. Unless the mapping file is upgraded, the new field will not be exchanged between ObjectServers.

Table list

alerts.status

alerts.journal

alerts.details

iduc_system.iduc_stats

security.users

security.groups

security.roles

security.role_grants

security.group_members

catalog.restrictions

security.restriction_filters

security.permissions

tools.menus

tools.menu_items

tools.actions

tools.action_access

tools.menu_defs

tools.prompt_defs

alerts.conversions

alerts.col_visuals
alerts.colors
master.servergroups
alerts.gsmaRIdCS
alerts.gsmaRIdCSRelationship
alerts.gsmaErrors
alerts.itm_situation_timeouts

Table 1 List of synchronized tables

9.1.1 Pre/Co Requisites

This is very important that fields defined in alert.status table are **totally** the same between both primary and backup Object Servers including the “**Ordinal Position**” parameter. That means that when new fields are added to the ObjectServer, they must be added in both Object-Server following the same order to keep this “Ordinal Position” synchronized between both Object Servers.

So, in upgrade procedure, the most important point to check is the creation of new fields. This concerns the upgrade of GSMA code as well as the upgrade of the Object Server itself.

Note that when a new field will be added in Object Server then a new Omnibus failover package will be delivered in order to include the new field in the list of the synchronized fields.

9.1.2 Upgrade GSMA EDM and Related Automation on Object-Server

Follow these steps:

1. Logon ncosys user on **Backup** Object Server.
2. Stop the aggregation gateway using Process Agent
\$OMNIHOME/bin/nco_pa_stop -process AggregationGateway
3. Upgrade GSMA Omnibus code of backup ObjectServer as described in the GSMA Release History document (Object Server and Common/shared Utilities codes)
4. Check there is no error.
5. Logon ncosys user on **Primary** Object Server.
6. Upgrade GSMA Omnibus code of primary ObjectServer as described in the GSMA Release History document. You must follow the exact same order you ran the nco_sql instructions on Backup Object Server in order to add the eventual new columns in same order.
7. Check there is no error.
8. Logon ncosys on **Backup** Object Server
9. Upgrade GSMA Failover code for Omnibus as described in the GSMA Omnibus Failover Release History document.
10. Restart the aggregation gateway using Process Agent
\$OMNIHOME/bin/nco_pa_start -process AggregationGateway

11. Check there is no error.

9.2 Upgrade GSMA Failover code for Aggregation gateway

Occasionally when changes have been made on table and fields a new failover package is delivered and a new release history document for the failover package is delivered as well indicating the procedure to upgrade the failover environment.

9.3 Upgrade GSMA EIF Probes code in failover mode

Follow these steps:

1. Logon ncosys user on **Primary** EIF probe system.
2. Download and upgrade the GSMA EIF probe code as described in the EIFPRB Release History document.
3. Verify the rules syntax with the syntax checker
/opt/IBM/GSMA/probe/EIF/bin/rules_check.ksh
Make sure you see "Rules file syntax OK" or correct the problem.
4. Now have the EIF probe reload the rules to pick up the config change
/opt/IBM/GSMA/probe/EIF/bin/eif_hup.ksh
5. Logon ncosys user on **Backup** EIF probe system.
6. Download and upgrade the GSMA EIF probe code as described in the EIFPRB Release History document.
7. Verify the rules syntax with the syntax checker
/opt/IBM/GSMA/probe/EIF/bin/rules_check.ksh
Make sure you see "Rules file syntax OK" or correct the problem
8. Now have the EIF probe reload the rules to pick up the config change
/opt/IBM/GSMA/probe/EIF/bin/eif_hup.ksh

Repeat steps 1 to 4 for all probes involved in your environment.

9.3.1 Error messages in Gateway log

The message "ObjectServerA: Valid connection not held by writer." indicates the writer which writes to PRINCOMS is not connected. From the logs, this is being received on gateway startup when the writer has not yet connected or on shutdown when the writer has already disconnected.

These errors can be safely ignored. However, you will also see this error if the gateway gets disconnected from the Object Server. In that case, additional errors would be logged on the connection being marked dead and the connection being closed. If you see those errors as well, we would need to look at the Object Server to see why the connection was lost. Please let us know if you have any questions.

9.3.2 Error messages in Backup Object Server log

The message Error: E-OBJ-102-016: Language command from impact@gsmevtxlp01.lagaude.ibm.com failed. Object not found on line 1 of statement 'select IsActingPrimary from master.impact_failback;' at or near 'impact_failback' can appears in Backup Object Server log.

These errors can be ignored. It refers to the Impact failback processing included in Impact 4.x versions that do not more apply.

Appendix A. List of default OMNibus triggers used during failover

Here is the list of default Omnibus triggers involved in the ObjectServer failover mechanism.

WARNING: do not disable them, neither modify them either on the Primary or Secondary ObjectServer.

Trigger	Comment	ObjectServer Property setting checked before execution
backup_startup	At signal start-up reception, disable automation triggers (triggers whose group is primary_only).	BackupObjectServer=TRUE
backup_counterpart_down	Enable automation triggers (triggers whose group is primary_only) when gw_counterpart_down signal is received from Aggregation gateway meaning the Primary ObjectServer is down. BackupObjectServer change property ActingPrimary to TRUE.	BackupObjectServer=TRUE
backup_counterpart_up	Disable automation triggers (triggers whose group is primary_only) when gw_counterpart_up signal is received from Aggregation gateway.	BackupObjectServer=TRUE
disconnect_all_clients	After each synchronization successful signal reception (gw_resync_finish), disconnect all clients other than Administrators, WEBTOP and failover gateways.	BackupObjectServer=TRUE