



5410 Presence Server

vPS 20.0.0 MTU Size Change Guide

Version: 1.0
WebLib ID: 3MT-42306-0005-SEZZA
Date: 08/25/2023

Table of Contents

1	INTRODUCTION	3
1.1	OVERVIEW	3
1.2	ON WHAT SYSTEMS SHOULD THIS PROCEDURE BE APPLIED?	3
1.3	HOW TO DETERMINE IF THE MTU SIZE CHANGE IS WORKING	3
1.4	OBTAIN SOFTWARE NEEDED	3
1.5	PROCEDURE TIMING, SEQUENCE, AND IMPACTS	3
1.6	GUIDE TO DOCUMENT	4
1.7	ACRONYMS	4
1.8	PREREQUISITES TO RUNNING THIS PROCEDURE	5
1.9	DOCUMENT HISTORY	5
2	PROBLEM(S) FIXED	6
3	BEFORE YOU BEGIN	7
3.1	REQUIRED SOFTWARE	7
3.2	REQUIRED HARDWARE	7
3.3	REQUIRED DATA	7
3.4	REQUIRED DOCUMENTS	7
3.5	REQUIRED PRE-WORK	7
3.6	CUSTOMER RESOURCES	8
4	PRE-WORK	9
4.1	OBTAINING USER PASSWORDS	9
4.2	CAPTURING A LOG OF THE PROCEDURE	9
4.3	ACCESS ASR GUI	9
4.3.1	Logging into ASR GUI	9
4.3.2	Logging into WebMT	9
4.4	TURN OFF THE DEBUG LOG	9
4.5	STOP PROVISIONING	10
4.6	GENERAL HEALTH CHECK	10
4.6.1	HELPME platform	10
4.6.2	Alarms	11
4.7	VERIFY THAT THE SYSTEM IS ON PS 20.0.0 OR LATER	11
4.8	BACKING UP THE SYSTEM	11
5	PROCEDURE	12
5.1	PRE-WORK	12
5.2	MAKE PILOT-A (0-0-1) THE ACTIVE PILOT	12
5.3	APPLY MTU SIZE CHANGE ON PS AND CASS VNF	13
5.4	vPS ONLY: RESTART ALL THE AGENTS FROM ASR GUI	22
5.5	POST-WORK	23
5.5.1	Verify System State	23
5.6	TEST	24
6	BACKOUT	25

1 Introduction

1.1 Overview

This document contains the procedure to change the MTU size on vPS FE and Cass DB VNF interfaces. Provisioning to this site should be halted during this procedure.

1.2 On what systems should this procedure be applied?

This procedure applies to any vPS Release on 20.0.0.904 or later.

1.3 How to determine if the MTU size change is working

1. Provisioning is successful.
2. Basic call processing is successful.
3. The MTU size is persistent:
 - a. upon rebooting any VM where the MTU size is updated.
 - b. upon switchovers of Pilot and IOH VM.
 - c. upon Heal and Heal force a rebuild.

1.4 Obtain Software Needed

See section 3.1 Required Software.

1.5 Procedure Timing, Sequence, and Impacts

The table below describes the timing, sequence, and service impacts of this procedure for a single site. This procedure needs to be run on both sites of the GR pair, but this does not need to happen in the same maintenance window.

Note: In the table below, **lightly shaded** cells indicate when there is a partial service impact. **Dark-shaded** cells indicate when system services are unavailable. Times are taken from the lab system. Times on the production system may vary.

PRE-WORK ACTIVITIES DAYS/WEEKS IN ADVANCE NO SERVICE IMPACT	MTCE WINDOW 1 SINGLE 5-HOUR MTCE WINDOW SERVICE IMPACT
Verify Presence Server - obtain password - health check - alarms - backup TIMING: 65 Min	Check pre-work TIMING: 15 Min
	Apply MTU size change script TIMING: 15 Min
	Post Work TIMING: 15 Min

	Testing
	TIMING: 20 Min

1.6 Guide to Document

Users are assumed to be familiar with the Presence Server and mCAS OA&M facilities. For example, users are familiar with the WebMT, ASR GUI, subshl, and general Linux command line commands.

Most commands are executed by root and nxuser users. Make sure to follow which user to run the command. Commands with [nxuser:] heading are meant to be executed by nxuser. Commands with [root:#] heading are meant to be executed by the root user.

Within each section of this document, a common text format is used to organize the document. Understanding the fonts allows the user to identify needed information more efficiently. The text format for the document is the following.

The regular text uses black, normal, 10 point, Times New Roman font.

The noted text uses black, italic, 10 point, Times New Roman font.

A system command to enter uses bold, 9 point, courier new font.

Platform generated text uses 9 point, courier new font

Platform file text uses normal, 9 points, courier new font.

Platform edited file text uses bold, 9 point, courier new font.

Font Sample

Font Sample

Font Sample

Font Sample

Font Sample

Font Sample

1.7 Acronyms

The following table provides definitions for acronyms used in this procedure:

Acronyms	Definition
ASR	Application Server Runtime
CID	Compass ID
CPR	Cross-Pilot Restore
GRM	Geo-redundancy Manager
HTTP	Hyper-Text Transport Protocol
IAS	IMS Application Server
IO	Input/Output
IOH	IO Handler
IP	Internet Protocol
LAN	Local Area Network
LV	Logical Volume
mCAS	modular Converged Application Server
MTU	Maximum Transmission Unit
Non-pilot	mCAS application host
OA	Onboard Administrator
OAM	Operations Administration Management
PILOT	mCAS administrative host
PS	Presence Server
R-C-S	Rack-Chassis-Server
RMS	Rack mounted server
SU	Software Upgrade/Software Update
VIP	Virtual IP

XDMS	XML Database Management System
XML	Extensible Markup Language

1.8 Prerequisites to running this procedure

1. Review the document and steps before applying in case there are questions or concerns.
2. Prior to the maintenance window, to minimize the maintenance window and to verify the sanity of the system, execute the steps in the **Pre-Work** section.

1.9 Document History

The following table provides a history of releases of this document:

Version	Date	Description
1.0	08/25/2023	Initial version for the 20.0.0 release.

2 Problem(s) fixed

This MOP addresses reverting MTU sizes to desired values.

3 Before You Begin

3.1 Required Software

Transfer the MTU size change script to the active pilot

The MTU size change script needs to be transferred to the virtual PS and Cassandra VNF.

1. Log in to the active pilot as a **nxuser** (for PS frontend VNF) **or cassandra** (for Cass backend VNF) user and switch to the root user.
2. Transfer the **updateMTUSize.sh** script to the **/root** directory.
3. Verify the md5sum for the **updateMTUSize.sh** script.

Enter:

```
md5sum updateMTUSize.sh
```

Expected result:

```
8f1cf769816fda0c7c6ff53cc4f008f6 updateMTUSize.sh
```

Note: If the value returned does not match the MD5sum value from above then download again and re-verify the md5sum.

3.2 Required Hardware

None

3.3 Required Data

None

3.4 Required Documents

None

3.5 Required Pre-Work

This procedure requires that the following pre-work be completed. Failure to complete these steps would delay the procedure start date. Use the **Complete** column to indicate the date the item was completed.

The details of these pre-work items are found in the Pre-Work section. Most of these pre-work items can be done outside of the maintenance window. This can save time in case an issue is found.

Pre-Work Required	When?	Owner	Complete?
1. Obtain pilot root, nxuser, and ainet user passwords. Obtain OA ainet password.	Prior to running the procedure	NOKIA/Customer	
2. Know how to capture log to a file	Prior to running the procedure	NOKIA/Customer	

Pre-Work Required	When?	Owner	Complete?
3. Know how to access WebMT and ASR GUI	Prior to running the procedure	NOKIA/Customer	
4. Know how to determine active/standby Pilot	Prior to running the procedure	NOKIA/Customer	
5. Turn off the DEBUG log	Prior to running the procedure	NOKIA/Customer	
6. General health check	Prior to running the procedure	NOKIA/Customer	
7. Checking the state of the ASR web GUI	Prior to running the procedure	NOKIA/Customer	
8. Backup of the system	Prior to running the procedure	NOKIA/Customer	

3.6 Customer Resources

This procedure requires the following customer resources. Ensure that the resources are available and informed prior to the procedure starting. Use the **Complete** column to indicate the date the resource was informed/aligned with the need.

Customer Resources Required	Complete?
The team that supports PS installation must be on-call during the maintenance window	

4 Pre-Work



This section takes about 15 minutes.

4.1 Obtaining User Passwords

The procedure requires logging into the system to execute commands. These user IDs and their associated passwords are needed.

Server	User Id	Password	Comments
Pilot	root		The root user is used to execute mCAS commands.
	nxuser		nxuser user is used to execute ASR commands and ASR GUI.
	ainet		ainet user is used in WebMT.
	cassandra		Used for Cassandra DB VNF login

4.2 Capturing a log of the procedure

It is important to capture the procedure output in a log file. Knowing how to capture a log to file depends on the type of terminal emulator. Make sure that you know how to capture a log to file. In case of a problem, the log file is invaluable to troubleshoot the issue.

Note: To capture the log, the installer must be able to access the system via serial console or direct ssh connection to the system.

4.3 Access ASR GUI

The subsections below provide details on how to log in to the WebMT and ASR GUI.

4.3.1 Logging into ASR GUI

1. Open a web browser and enter the URL of the OA&M Virtual IP address of the PS system with port number 8443 (for example: **https://<OAM VIP address>:8443**).
2. Log in using the ASR admin account **nxuser** and password.
3. On the left side of the page, select **Deploy/Configure installed applications**.

4.3.2 Logging into WebMT

1. Open a web browser and enter the URL of the OA&M Virtual IP address of the PS system with no port number (for example: **https://<OAM VIP address>**).
2. Log in using the WebMT admin account **ainet** and applicable password.

4.4 Turn off the DEBUG Log

This step is to verify (and turn off if necessary) all DEBUG log messages for all callout agents.

1. Log in to the ASR GUI.

2. Go to the **Group Deployment** tab.
3. For each CalloutAgent (second column), select **Extensive View** -> **General Parameters** -> **Logging Parameters**.
4. Verify that the **Default level** is WARN. If it is not, set it to WARN.
5. Verify that the **General Destination(s)** is SYSLOGNG or RSYSLOG. If it is not, set it to RSYSLOG.
6. If you make any changes, click the **Apply** button and then the **Back** button near the upper right corner. Click on **Apply Changes** to apply the changes.
7. Repeat for all other callout agents.

4.5 Stop Provisioning

To prevent provisioning, stop the AggXcapProv agent on the PS VNF.

1. Log in to the ASR GUI. For example, <https://<OAM VIP>:8443>.
2. Click on the **Deploy/Configure installed applications**.
3. Under the Group Control page, click on the **AggXcapProv** agent group as shown below.

Start

STOP

By Host

☐ Select/Unselect all groups

Group	Proxy component	Host	Instance	Application	Monitored proxylet
☐ System	FastCache	asr-1 (psblade1-0-0-1)	☐ master		
		asr-2 (psblade1-0-0-9)	☐ slave		

Group	Proxy component	Host	Instance	Application	Monitored proxylet
☒ AggXcapProv	CalloutAgent	asr-1 (psblade1-0-0-1)	☒ asr_1_callout_agent_axp1	+	
		asr-2 (psblade1-0-0-9)	☒ asr_2_callout_agent_axp2	+	
	Http Stack	asr-1 (psblade1-0-0-1)	☒ asr_1_http_stack_axp		
		asr-2 (psblade1-0-0-9)	☒ asr_2_http_stack_axp		

4. Click on the **STOP** button to stop the agent.
5. Click on the **Yes** button when prompted.

Warning: This will shutdown this application

Are you sure you want to proceed ?

Yes **No**

4.6 General Health Check

4.6.1 HELPME platform

1. Log in to the active pilot as a root user and run the following command:

```
[root:#] HELPME platform
```

2. Log in to the active pilot as a root user and run the following command:

```
[root:#] HELPME cassandra
```

3. Correct (or justify) any errors/warnings.

Consult with Nokia support if assistance is needed to resolve issues identified by HELPME.

4.6.2 Alarms

Check and verify the system does not have any alarms that need to be resolved before running this procedure. Do this by logging into the ASR GUI, clicking on the ALARMS MANAGER tab, and sorting the alarms by Severity (click the down arrow on the Severity column) from highest to lowest.

If there are any alarms, either resolve those alarms, or escalate to your next level of support.

4.7 Verify that the system is on PS 20.0.0 or later

1. Log in to vPS, as a nxuser on the active pilot, and execute the following command:

```
[nxuser:#] /opt/proxy/scripts/admin/ppversion.sh
```

Expected output for PS 20.0.0.904:

```
Geo Redundancy: 6.4.107.904
IMS Presence Server: 20.0.0.904
Application Server Runtime: 6.4.107.904
PS scripts for MySQL: 20.0.0.904
Xdm Server: 20.0.0.904
XDMS scripts for MySQL: 20.0.0.904
```

2. Verify that the IMS Presence Server release is as shown in the yellow highlight in the output above.

4.8 Backing up the System

Before the procedure is executed, the system should be backed up (using whatever customer-specific backup procedures have been defined) so that in case of a disaster there will be a safe copy that can be used to restore the system to its original state.

5 Procedure

5.1 Pre-Work

Ensure that any work identified in the **Before You Begin** section has been completed prior to the start of this maintenance procedure.

5.2 Make Pilot-A (0-0-1) the Active Pilot

This section takes about 15 minutes if the pilot switch-over is required.

Using WebMT or subshl, verify that **Pilot-A** is the active pilot. If **Pilot-A** is not the active pilot, then execute a pilot switch over with the procedure below. Make sure all systems have recovered and are ready before proceeding to the next section. All instructions from this point rely on the fact that **Pilot-A** is the active pilot, so be sure that **Pilot-A** is the active pilot and **Pilot-B** is a standby pilot before proceeding.

1. Connect to the **active pilot** as a **root** user.

Note: Make Pilot-A active if it is not active.

2. Check the current state of the Pilots.

Enter:

```
[root:#] subshl
```

Enter:

```
op:status,machine=all;
```

Sample Output:

MACHINE	STATE	REASON	INIT STATUS	SOFT CHK	NODE ID	VHOST NAME	VHOST STATE
adm001	LEAD	AUTO	COMPL	ALW		pilot	ACTIVE
adm002	LEAD	AUTO	COMPL	ALW		pilot	STANDBY
ioh001	LEAD	AUTO	COMPL	ALW		vhost-1	STANDBY
ioh002	LEAD	AUTO	COMPL	ALW		vhost-1	ACTIVE
agt001	LEAD	AUTO	COMPL	ALW			
agt002	LEAD	AUTO	COMPL	ALW			

If **Pilot-A** is the active pilot, quit the **subshl** and skip to the next section.

Enter:

```
quit;
```

If **Pilot-A** is NOT the active pilot, execute a pilot switchover using the following commands on **Pilot-B** as **root**.

Enter:

```
SW:PILOT;
```

Sample Output:

```
SW:PILOT WILL AFFECT CALL PROCESSING, CONTINUE Y OR N? y; PF
+++ PLANOMCAS002 2012-02-29 14:14:50 MAINT /IN014 #000833 0-0-9 >
M SW PILOT REQUEST SUCCESSFULLY ACKNOWLEDGED
END OF REPORT #000833+-
< Connection to <system name> closed.
```

Note: This action closes the connection and reboots Pilot-B. Pilot-A becomes the active pilot. Wait for some time for the Pilot-A status to change to active.

- On the **active pilot (Pilot-A)** as **root**, execute:

Enter:

```
[root:#] subsh1
```

Enter:

```
op:status,machine=all;
```

Sample Output:

MACHINE	STATE	REASON	INIT STATUS	SOFT CHK	NODE ID	VHOST NAME	VHOST STATE
adm001	LEAD	AUTO	COMPL	ALW		pilot	ACTIVE
adm002	LEAD	AUTO	COMPL	ALW		pilot	STANDBY
ioh001	LEAD	AUTO	COMPL	ALW		vhost-1	STANDBY
ioh002	LEAD	AUTO	COMPL	ALW		vhost-1	ACTIVE
agt001	LEAD	AUTO	COMPL	ALW			
agt002	LEAD	AUTO	COMPL	ALW			

Enter:

```
quit;
```

5.3 Apply MTU size change on PS and Cass VNF



The timing of this section takes about 10 minutes on a PS FRONTEND VNF and CASSANDRA DB VNF.

- Go to the script directory on the active pilot as a root user.

```
[root:#] cd /root
[root:#] chmod +x updateMTUSize.sh
```

2. Take the **ifconfig** output of the interface where the MTU size is about to be changed and compare the same after running the procedure for validation.

As a root user on the active pilot, run the following commands.

[root:#] ifconfig

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9050
      inet 172.16.64.1 netmask 255.255.0.0 broadcast 172.16.255.255
      inet6 fe80::f816:3eff:fe2c:d65a prefixlen 64 scopeid 0x20<link>
      ether fa:16:3e:2c:d6:5a txqueuelen 1000 (Ethernet)
      RX packets 41928908 bytes 5217068367 (4.8 GiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 44393016 bytes 12000358744 (11.1 GiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9050
      inet6 fe80::f816:3eff:fedf:75a9 prefixlen 64 scopeid 0x20<link>
      ether fa:16:3e:df:75:a9 txqueuelen 1000 (Ethernet)
      RX packets 20958 bytes 3873713 (3.6 MiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 15838 bytes 4516956 (4.3 MiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 9050
      inet6 fe80::f816:3eff:fe16:bdfd prefixlen 64 scopeid 0x20<link>
      inet6 2001:470:eelf:755::13 prefixlen 64 scopeid 0x0<global>
      inet6 2001:470:eelf:755::11 prefixlen 64 scopeid 0x0<global>
      ether fa:16:3e:16:bd:fd txqueuelen 1000 (Ethernet)
      RX packets 2856197 bytes 2612206143 (2.4 GiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 3062390 bytes 200652605 (191.3 MiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

3. On the active pilot as a root user specify the interface name and new MTU size as the argument to execute the script successfully. Take note of the current MTU size before changing the MTU, this will help in case backout is done later.

Syntax to execute the script:

[root:#] ./updateMTUSize.sh -i <interface name> -m <new MTU size>

Note: The script can take 4 options as arguments and the log of the above command will be stored on **/opt/config/log/updateMTUSize.sh.out**

-h|--help|--help: To be used to view script description.

-i: To collect the interface name from the user.

-m: To collect the MTU size from the user.

-b: Backs out to the previous MTU size, default value is set as 0 (false). Refer Backout Section.

Example of MTU size changes from 9050 to 1500 on the db6 interface below:

```
psIIpstack6-adm001:/root-# ./updateMTUSize.sh -i db6 -m 1500
```

```
02:49:52 updateMTUSize.sh:
```

```
02:49:52 updateMTUSize.sh: ===== adm001 (0-0-1) =====
02:49:52 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:49:53 updateMTUSize.sh:
##### MTU value before change #####
02:49:53 updateMTUSize.sh:      MTU=9050
02:49:53 updateMTUSize.sh:
##### MTU value after change #####
02:49:53 updateMTUSize.sh:      MTU=1500
02:49:53 updateMTUSize.sh: For dba/dba6 interfaces MTU change to
reflect, the dba interface must be brought down and up.
02:49:53 updateMTUSize.sh:
```

```
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dba not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dba6 not found.
Usage: ifup <configuration>
02:49:57 updateMTUSize.sh: Ignore the above info "ifcfg-dba6(A) not
found". This is expected when the interface is not present on a
particular VM
02:49:57 updateMTUSize.sh:
```

```
02:49:57 updateMTUSize.sh: ===== adm002 (0-0-2) =====
02:49:57 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:49:57 updateMTUSize.sh:
##### MTU value before change #####
02:49:57 updateMTUSize.sh:      MTU=9050
02:49:58 updateMTUSize.sh:
##### MTU value after change #####
02:49:58 updateMTUSize.sh:      MTU=1500
02:49:58 updateMTUSize.sh: For dba/dba6 interfaces MTU change to
reflect, the dba interface must be brought down and up.
02:49:58 updateMTUSize.sh:
```

```
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dba not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dba6 not found.
Usage: ifup <configuration>
02:50:02 updateMTUSize.sh: Ignore the above info "ifcfg-dba6(A) not
found". This is expected when the interface is not present on a
particular VM
02:50:02 updateMTUSize.sh:
```

```
02:50:02 updateMTUSize.sh: ===== ioh001 (0-0-3) =====
02:50:02 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:50:02 updateMTUSize.sh:
##### MTU value before change #####
02:50:02 updateMTUSize.sh:      MTU=9050
02:50:02 updateMTUSize.sh:
##### MTU value after change #####
02:50:02 updateMTUSize.sh:      MTU=1500
02:50:03 updateMTUSize.sh: For dba/dba6 interfaces MTU change to
reflect, the dba interface must be brought down and up.
02:50:03 updateMTUSize.sh:

ERROR      : [ipv6_test_device_status] Missing parameter 'device' (arg 1)
ERROR      : [/etc/sysconfig/network-scripts/ifup-eth] Device name does
not seem to be present.
/usr/sbin/ifup: configuration for ifcfg-dba not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dba6 not found.
Usage: ifup <configuration>
02:50:04 updateMTUSize.sh: Ignore the above info "ifcfg-dba6(A) not
found". This is expected when the interface is not present on a
particular VM
02:50:04 updateMTUSize.sh:

02:50:04 updateMTUSize.sh: ===== ioh002 (0-0-4) =====
02:50:04 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:50:04 updateMTUSize.sh:
##### MTU value before change #####
02:50:04 updateMTUSize.sh:      MTU=9050
02:50:05 updateMTUSize.sh:
##### MTU value after change #####
02:50:05 updateMTUSize.sh:      MTU=1500
02:50:05 updateMTUSize.sh: For dba/dba6 interfaces MTU change to
reflect, the dba interface must be brought down and up.
02:50:05 updateMTUSize.sh:

ERROR      : [ipv6_test_device_status] Missing parameter 'device' (arg 1)
ERROR      : [/etc/sysconfig/network-scripts/ifup-eth] Device name does
not seem to be present.
/usr/sbin/ifup: configuration for ifcfg-dba not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dba6 not found.
Usage: ifup <configuration>
02:50:06 updateMTUSize.sh: Ignore the above info "ifcfg-dba6(A) not
found". This is expected when the interface is not present on a
particular VM
02:50:06 updateMTUSize.sh:
```

```
02:50:06 updateMTUSize.sh: ===== agt001 (0-0-5) =====
02:50:06 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:50:06 updateMTUSize.sh:
##### MTU value before change #####
02:50:07 updateMTUSize.sh:      MTU=9050
02:50:07 updateMTUSize.sh:
##### MTU value after change #####
02:50:07 updateMTUSize.sh:      MTU=1500
02:50:07 updateMTUSize.sh: For dba/dba6 interfaces MTU change to
reflect, the dba interface must be brought down and up.
02:50:07 updateMTUSize.sh:

INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dba not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dba6 not found.
Usage: ifup <configuration>
02:50:11 updateMTUSize.sh: Ignore the above info "ifcfg-dba6(A) not
found". This is expected when the interface is not present on a
particular VM
02:50:11 updateMTUSize.sh:

02:50:11 updateMTUSize.sh: ===== agt002 (0-0-6) =====
02:50:11 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:50:11 updateMTUSize.sh:
##### MTU value before change #####
02:50:11 updateMTUSize.sh:      MTU=9050
02:50:12 updateMTUSize.sh:
##### MTU value after change #####
02:50:12 updateMTUSize.sh:      MTU=1500
02:50:12 updateMTUSize.sh: For dba/dba6 interfaces MTU change to
reflect, the dba interface must be brought down and up.
02:50:12 updateMTUSize.sh:

INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dba not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dba6 not found.
Usage: ifup <configuration>
```

```
02:50:16 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:50:16 updateMTUSize.sh:
```

```
02:50:16 updateMTUSize.sh: ===== adm001 (0-0-1) =====
02:50:16 updateMTUSize.sh:
##### MTU value before change #####
02:50:16 updateMTUSize.sh:      MTU=9050
02:50:16 updateMTUSize.sh:
##### MTU value after change #####
02:50:16 updateMTUSize.sh:      MTU=1500
02:50:16 updateMTUSize.sh: For db6/dbe6 interfaces MTU change to
reflect, the db6A interface must be brought down and up.
02:50:16 updateMTUSize.sh:
```

```
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:50:20 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:50:20 updateMTUSize.sh:
```

```
02:50:20 updateMTUSize.sh: ===== adm002 (0-0-2) =====
02:50:20 updateMTUSize.sh:
##### MTU value before change #####
02:50:20 updateMTUSize.sh:      MTU=9050
02:50:20 updateMTUSize.sh:
##### MTU value after change #####
02:50:20 updateMTUSize.sh:      MTU=1500
02:50:20 updateMTUSize.sh: For db6/dbe6 interfaces MTU change to
reflect, the db6A interface must be brought down and up.
02:50:20 updateMTUSize.sh:
```

```
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
```

```
02:50:24 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
```

```
02:50:24 updateMTUSize.sh:
```

```
02:50:24 updateMTUSize.sh: ===== ioh001 (0-0-3) =====
02:50:24 updateMTUSize.sh: Entering to Non-Pilot Network Configurations
02:50:24 updateMTUSize.sh: There is no interface present as db6 on
ioh001( 0-0-3 )
02:50:24 updateMTUSize.sh:
```

```
02:50:24 updateMTUSize.sh: ===== ioh002 (0-0-4) =====
02:50:24 updateMTUSize.sh: Entering to Non-Pilot Network Configurations
02:50:24 updateMTUSize.sh: There is no interface present as db6 on
ioh002( 0-0-4 )
02:50:25 updateMTUSize.sh:
```

```
02:50:25 updateMTUSize.sh: ===== agt001 (0-0-5) =====
02:50:25 updateMTUSize.sh: Entering to Non-Pilot Network Configurations
02:50:25 updateMTUSize.sh:
##### MTU value before change #####
02:50:25 updateMTUSize.sh:          MTU=9050
02:50:25 updateMTUSize.sh:
##### MTU value after change #####
02:50:25 updateMTUSize.sh:          MTU=1500
02:50:25 updateMTUSize.sh: For db6/dbe6 interfaces MTU change to
reflect, the db6A interface must be brought down and up.
02:50:25 updateMTUSize.sh:
```

```
INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:50:29 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:50:29 updateMTUSize.sh:
```

```
02:50:29 updateMTUSize.sh: ===== agt002 (0-0-6) =====
```

```

02:50:29 updateMTUSize.sh: Entering to Non-Pilot Network Configurations
02:50:29 updateMTUSize.sh:
##### MTU value before change #####
02:50:29 updateMTUSize.sh:      MTU=9050
02:50:29 updateMTUSize.sh:
##### MTU value after change #####
02:50:29 updateMTUSize.sh:      MTU=1500
02:50:29 updateMTUSize.sh: For db6/dbe6 interfaces MTU change to
reflect, the db6A interface must be brought down and up.
02:50:29 updateMTUSize.sh:

INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:50:32 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:50:32 updateMTUSize.sh: MTU update Logs file is stored as
/opt/config/log/updateMTUSize.sh.out

```

Note: Ignore any error that occurs similar to the above output. Contact support for more clarifications.

4. The script (updateMTUSize.sh) does not support rebooting the oam* interfaces after changing their MTU size. For the updated MTU size to take effect, follow the below procedures as root on the active pilot:

- a. After the execution of the script, run the below command from the active pilot.

```

[root:#] ifdown ifcfg-oam
[root:#] ifdown ifcfg-oamA

```

Note: After bringing down the oam* interface, the ssh connection to the active VIP will be dropped. However, this will not switchover the Pilots.

- b. Since the pilot VIP is down, access the active pilot from the standby pilot. Log in directly to the standby pilot OAM IP as a nxuser and become a root user.

```

[root:#] ssh <standby pilot's OAM IP>

```

Log in to the active pilot from the standby pilot.

```

[root:#] ssh 0-0-1

```

- c. Bring up the oam interfaces on the active pilot :

```

[root:#] ifup ifcfg-oam
[root:#] ifup ifcfg-oamA
[root:#] exit
[root:#] exit

```

- d. After the active pilot is up, log in to the pilot OAM VIP as a nxuser and become a root user.

```

[root:#] ssh <Active Pilot OAM VIP>

```

Log in to the standby pilot from the active pilot.

```
[root:#] ssh 0-0-2
```

Restart the standby pilot OAM by executing the below:

```
[root:#] ifdown ifcfg-oam
```

```
[root:#] ifup ifcfg-oam
```

5. As a root user from the active pilot, run the below command to verify the changes. Note that these changes will be persistent following a reboot or heal.

Note: Below is the example output taken after changing all the interfaces to 1500.

```
[root:#] ifconfig
```

```
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet 192.168.64.1 netmask 255.255.0.0 broadcast
192.168.255.255
      inet6 fe80::f816:3eff:fed8:501e prefixlen 64 scopeid
0x20<link>
      ether fa:16:3e:d8:50:1e txqueuelen 1000 (Ethernet)
      RX packets 265401327 bytes 27687983082 (25.7 GiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 288560061 bytes 51587649470 (48.0 GiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth1: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet6 fe80::f816:3eff:fedd:8276 prefixlen 64 scopeid
0x20<link>
      ether fa:16:3e:dd:82:76 txqueuelen 1000 (Ethernet)
      RX packets 123470 bytes 21175817 (20.1 MiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 113821 bytes 34515817 (32.9 MiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth2: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet6 fe80::f816:3eff:fedf:1ada prefixlen 64 scopeid
0x20<link>
      inet6 2001:470:eelf:655::5001 prefixlen 64 scopeid 0x0<global>
      ether fa:16:3e:df:1a:da txqueuelen 1000 (Ethernet)
      RX packets 275878 bytes 70605042 (67.3 MiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 407891 bytes 32074959 (30.5 MiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
      inet6 fe80::f816:3eff:fec5:86eb prefixlen 64 scopeid
0x20<link>
      inet6 2001:470:eelf:651::5003 prefixlen 64 scopeid 0x0<global>
      inet6 2001:470:eelf:651::5001 prefixlen 64 scopeid 0x0<global>
      ether fa:16:3e:c5:86:eb txqueuelen 1000 (Ethernet)
      RX packets 12355 bytes 665782 (650.1 KiB)
      RX errors 0 dropped 0 overruns 0 frame 0
      TX packets 2704 bytes 232868 (227.4 KiB)
      TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

```

eth0:SYSTEMA: flags=4163<UP,BROADCAST,RUNNING, MULTICAST> mtu 1500
    inet 192.168.64.100 netmask 255.255.0.0 broadcast
    192.168.255.255
    ether fa:16:3e:d8:50:1e txqueuelen 1000 (Ethernet)
eth1:oam: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.40.150.17 netmask 255.255.255.0 broadcast
    10.40.150.255
    ether fa:16:3e:dd:82:76 txqueuelen 1000 (Ethernet)

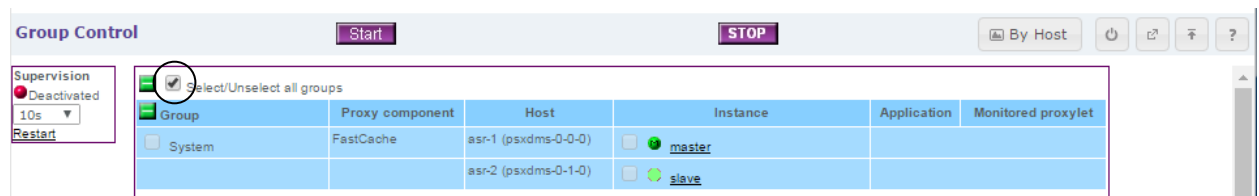
eth1:oamA: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.40.150.19 netmask 255.255.255.0 broadcast
    10.40.150.255
    ether fa:16:3e:dd:82:76 txqueuelen 1000 (Ethernet)

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
RX packets 69865585 bytes 13358687244 (12.4 GiB)
RX errors 0 dropped 0 overruns 0 frame 0
TX packets 69865585 bytes 13358687244 (12.4 GiB)
TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

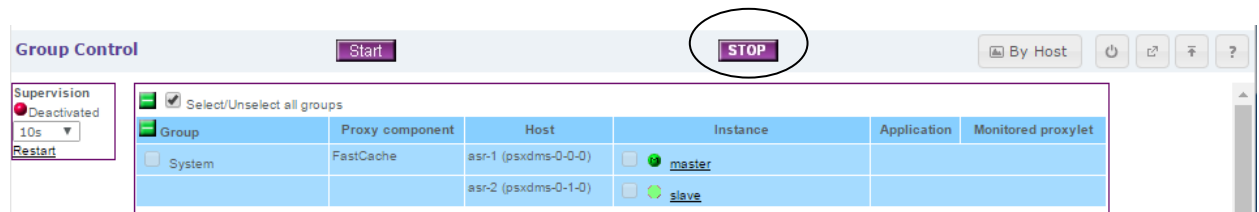
```

5.4 vPS only: Restart all the agents from ASR GUI.

1. Log in to the PS system as a **nxuser**.
2. Click on **Deploy/Configure installed applications**.
3. To stop all agents except the Locator group, go to the **Group Control** tab and select all agents by checking the Select/Unselect all groups as shown below.



4. Click on the **STOP** button to stop all the agents.



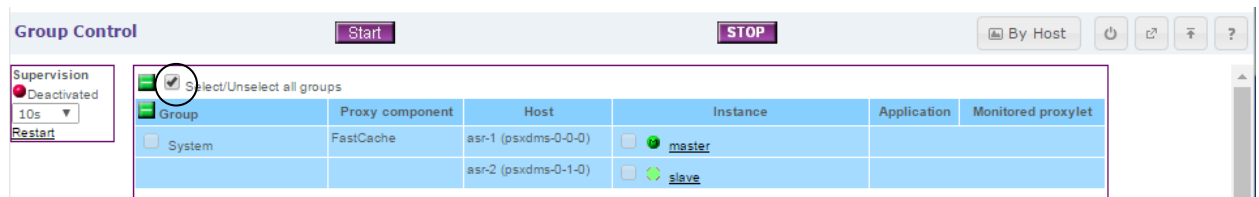
- Click on the **Yes** button when prompted.

Warning: This will shutdown this application

Are you sure you want to proceed ?

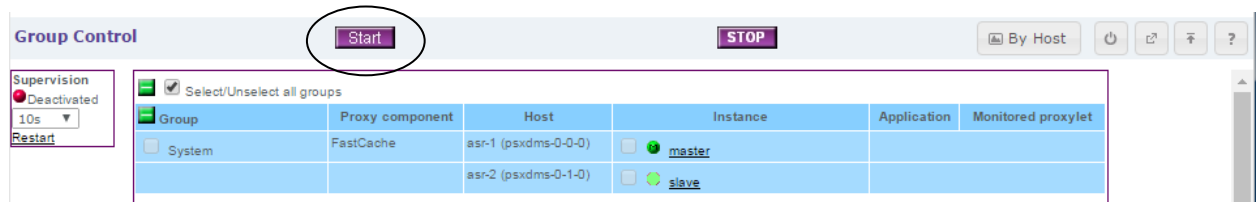
Yes No

- Wait for all the agents to stop.
- To start all agents, go to the **Group Control** tab and select all agents by checking **Select/Unselect all groups** as shown below.



- Click on the **Start** button to start all the agents.

Note: Do not click on the Start button more than once!



Wait until all the agents are started. You may want to click on the **Group Control** tab to refresh the page.

5.5 Post-Work

5.5.1 Verify System State

This section ensures the system state is returned to normal.

HELPME platform

- Log in to the active pilot as a root user of PS frontend VNF and run the following command:

```
[root:~] # HELPME platform
```

- Log in to the active pilot as a root user of a cassandra VNF and run the following command:

```
[root:~] # HELPME cassandra
```

- Correct (or justify) any errors/warnings.

Consult with Nokia support if assistance is needed to resolve issues identified by HELPME.

5.6 Test

This section ensures the system is operating as expected.



The timing of this section is customer-dependent but takes about 15 minutes.

1. Run initial PUBLISH from Presentity and SUBSCRIBE from Watcher. The expected result is NOTIFY contains the presentity's Document.
2. If any test fails, **follow the normal escalation procedure.**

6 Backout

The backout to the previous MTU size is always possible by giving a '-b' option to the script. The '-b' option is a flag that can be either 0 (false) or 1 (true). It is set as false by default.

1. On the active pilot as a root user specify the interface name, old MTU size, and -b set to true as mentioned below :

```
[root:#] ./updateMTUSize.sh -i <interface name> -m <old MTU size> -b 1
```

Example of MTU size changes from 1500 to the previous MTU size 9050 on the dbe6 interface below:

```
psIIPstack6-adm001:/root-# ./updateMTUSize.sh -i dbe6 -m 9050 -b 1
02:48:33 updateMTUSize.sh:

02:48:33 updateMTUSize.sh: ===== adm001 (0-0-1) =====
02:48:33 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:48:34 updateMTUSize.sh:
##### MTU value before change #####
02:48:34 updateMTUSize.sh:      MTU=1500
02:48:34 updateMTUSize.sh:
##### MTU value after change #####
02:48:34 updateMTUSize.sh:      MTU=9050
02:48:34 updateMTUSize.sh: For dbe/dbe6 interfaces MTU change to
reflect, the dbeA interface must be brought down and up.
02:48:34 updateMTUSize.sh:

INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:48:38 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:48:38 updateMTUSize.sh:

02:48:38 updateMTUSize.sh: ===== adm002 (0-0-2) =====
02:48:38 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:48:38 updateMTUSize.sh:
##### MTU value before change #####
02:48:38 updateMTUSize.sh:      MTU=1500
02:48:39 updateMTUSize.sh:
##### MTU value after change #####
02:48:39 updateMTUSize.sh:      MTU=9050
```

```
02:48:39 updateMTUSize.sh: For dbe/dbe6 interfaces MTU change to
reflect, the dbeA interface must be brought down and up.
02:48:39 updateMTUSize.sh:
```

```
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:48:43 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:48:43 updateMTUSize.sh:
```

```
02:48:43 updateMTUSize.sh: ===== ioh001 (0-0-3) =====
02:48:43 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:48:43 updateMTUSize.sh:
##### MTU value before change #####
02:48:43 updateMTUSize.sh:      MTU=1500
02:48:43 updateMTUSize.sh:
##### MTU value after change #####
02:48:44 updateMTUSize.sh:      MTU=9050
02:48:44 updateMTUSize.sh: For dbe/dbe6 interfaces MTU change to
reflect, the dbeA interface must be brought down and up.
02:48:44 updateMTUSize.sh:
```

```
ERROR     : [ipv6_test_device_status] Missing parameter 'device' (arg 1)
ERROR     : [/etc/sysconfig/network-scripts/ifup-eth] Device name does
not seem to be present.
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:48:45 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:48:45 updateMTUSize.sh:
```

```
02:48:45 updateMTUSize.sh: ===== ioh002 (0-0-4) =====
02:48:45 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:48:45 updateMTUSize.sh:
##### MTU value before change #####
02:48:46 updateMTUSize.sh:      MTU=1500
02:48:46 updateMTUSize.sh:
```

```
##### MTU value after change #####
02:48:46 updateMTUSize.sh:      MTU=9050
02:48:46 updateMTUSize.sh: For dbe/dbe6 interfaces MTU change to
reflect, the dbeA interface must be brought down and up.
02:48:46 updateMTUSize.sh:

ERROR      : [ipv6_test_device_status] Missing parameter 'device' (arg 1)
ERROR      : [/etc/sysconfig/network-scripts/ifup-eth] Device name does
not seem to be present.
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:48:47 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:48:47 updateMTUSize.sh:

02:48:47 updateMTUSize.sh: ===== agt001 (0-0-5) =====
02:48:47 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:48:48 updateMTUSize.sh:
##### MTU value before change #####
02:48:48 updateMTUSize.sh:      MTU=1500
02:48:48 updateMTUSize.sh:
##### MTU value after change #####
02:48:48 updateMTUSize.sh:      MTU=9050
02:48:48 updateMTUSize.sh: For dbe/dbe6 interfaces MTU change to
reflect, the dbeA interface must be brought down and up.
02:48:48 updateMTUSize.sh:

INFO       : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
INFO       : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:48:52 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:48:52 updateMTUSize.sh:

02:48:52 updateMTUSize.sh: ===== agt002 (0-0-6) =====
02:48:52 updateMTUSize.sh: Verifying if the interface is present on the
asked interface
02:48:52 updateMTUSize.sh:
##### MTU value before change #####
```

```

02:48:53 updateMTUSize.sh:          MTU=1500
02:48:53 updateMTUSize.sh:
##### MTU value after change #####
02:48:53 updateMTUSize.sh:          MTU=9050
02:48:53 updateMTUSize.sh: For dbe/dbe6 interfaces MTU change to
reflect, the dbeA interface must be brought down and up.
02:48:53 updateMTUSize.sh:

INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:48:57 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:48:57 updateMTUSize.sh:

02:48:57 updateMTUSize.sh: ===== adm001 (0-0-1) =====
02:48:57 updateMTUSize.sh:
##### MTU value before change #####
02:48:57 updateMTUSize.sh:          MTU=1500
02:48:57 updateMTUSize.sh:
##### MTU value after change #####
02:48:57 updateMTUSize.sh:          MTU=9050
02:48:57 updateMTUSize.sh: For dbe/dbe6 interfaces MTU change to
reflect, the dbeA interface must be brought down and up.
02:48:57 updateMTUSize.sh:

INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:49:01 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:49:01 updateMTUSize.sh:

02:49:01 updateMTUSize.sh: ===== adm002 (0-0-2) =====
02:49:01 updateMTUSize.sh:
##### MTU value before change #####

```

```
02:49:01 updateMTUSize.sh:          MTU=1500
02:49:01 updateMTUSize.sh:
##### MTU value after change #####
02:49:01 updateMTUSize.sh:          MTU=9050
02:49:01 updateMTUSize.sh: For db6/dbe6 interfaces MTU change to
reflect, the db6A interface must be brought down and up.
02:49:01 updateMTUSize.sh:

INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth2 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-db6A not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-db6A not found.
Usage: ifup <configuration>
02:49:05 updateMTUSize.sh: Ignore the above info "ifcfg-db6(A) not
found". This is expected when the interface is not present on a
particular VM
02:49:05 updateMTUSize.sh:

02:49:05 updateMTUSize.sh: ===== ioh001 (0-0-3) =====
02:49:05 updateMTUSize.sh: Entering to Non-Pilot Network Configurations
02:49:05 updateMTUSize.sh: There is no interface present as db6 on
ioh001( 0-0-3 )
02:49:05 updateMTUSize.sh:

02:49:05 updateMTUSize.sh: ===== ioh002 (0-0-4) =====
02:49:05 updateMTUSize.sh: Entering to Non-Pilot Network Configurations
02:49:05 updateMTUSize.sh: There is no interface present as db6 on
ioh002( 0-0-4 )
02:49:05 updateMTUSize.sh:

02:49:05 updateMTUSize.sh: ===== agt001 (0-0-5) =====
02:49:05 updateMTUSize.sh: Entering to Non-Pilot Network Configurations
02:49:06 updateMTUSize.sh:
##### MTU value before change #####
02:49:06 updateMTUSize.sh:          MTU=1500
02:49:06 updateMTUSize.sh:
##### MTU value after change #####
02:49:06 updateMTUSize.sh:          MTU=9050
02:49:06 updateMTUSize.sh: For db6/dbe6 interfaces MTU change to
reflect, the db6A interface must be brought down and up.
02:49:06 updateMTUSize.sh:
```

```

INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:49:09 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:49:10 updateMTUSize.sh:

02:49:10 updateMTUSize.sh: ===== agt002 (0-0-6) =====
02:49:10 updateMTUSize.sh: Entering to Non-Pilot Network Configurations
02:49:10 updateMTUSize.sh:
##### MTU value before change #####
02:49:10 updateMTUSize.sh:      MTU=1500
02:49:10 updateMTUSize.sh:
##### MTU value after change #####
02:49:10 updateMTUSize.sh:      MTU=9050
02:49:10 updateMTUSize.sh: For dbe/dbe6 interfaces MTU change to
reflect, the dbeA interface must be brought down and up.
02:49:10 updateMTUSize.sh:

INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
INFO      : [ipv6_wait_tentative] Waiting for interface eth1 IPv6
address(es) to leave the 'tentative' state
RTNETLINK answers: File exists
/usr/sbin/ifup: configuration for ifcfg-dbeA not found.
Usage: ifup <configuration>
/usr/sbin/ifup: configuration for ifcfg-dbe6A not found.
Usage: ifup <configuration>
02:49:14 updateMTUSize.sh: Ignore the above info "ifcfg-dbe6(A) not
found". This is expected when the interface is not present on a
particular VM
02:49:14 updateMTUSize.sh: MTU update Logs file is stored as
/opt/config/log/updateMTUSize.sh.out

```

2. Execute the procedures in sections 5.3 (from step 4) to 5.6 to complete the backout.

END OF PROCEDURE