



## **Cisco ASR 5000 Series GTPP Storage Server Installation and Administration Guide**

**Version 9.0**

**Last Updated August 31, 2011**

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

Text Part Number: OL-25585-02

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The following information is for FCC compliance of Class A devices: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio-frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case users will be required to correct the interference at their own expense.

The following information is for FCC compliance of Class B devices: This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If the equipment causes interference to radio or television reception, which can be determined by turning the equipment off and on, users are encouraged to try to correct the interference by using one or more of the following measures:

Reorient or relocate the receiving antenna.

Increase the separation between the equipment and receiver.

Connect the equipment into an outlet on a circuit different from that to which the receiver is connected.

Consult the dealer or an experienced radio/TV technician for help.

Modifications to this product not authorized by Cisco could void the FCC approval and negate your authority to operate the product.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Cisco and the Cisco Logo are trademarks of Cisco Systems, Inc. and/or its affiliates in the U.S. and other countries. A listing of Cisco's trademarks can be found at [www.cisco.com/go/trademarks](http://www.cisco.com/go/trademarks). Third party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco ASR 5000 Series GTPP Storage Server Installation and Administration Guide

© 2011 Cisco Systems, Inc. and/or its affiliated entities. All rights reserved.

# CONTENTS

---

<b>About this Guide .....</b>	<b>V</b>
Conventions Used.....	vi
Contacting Customer Support .....	viii
<b>GTPP Storage Server Overview .....</b>	<b>9</b>
Product Description.....	10
Partnering with a GSN.....	10
System Requirements and Recommendations.....	11
Minimum System Requirements for Stand-alone Deployment .....	11
Minimum System Requirements for Cluster Deployment.....	11
Default Ports for GSS .....	12
GSS Hardware Sizing and Provisioning Guidelines.....	13
Hard Drive Partition Recommendations .....	13
IP Multipathing (IPMP) on GSS Server (Optional).....	14
Features of the GSS .....	15
GSS Server Application.....	15
PostgreSQL Database Engine 8.4.4 .....	15
GSS FileGen Utility.....	15
File Format Encoding for CDRs .....	15
Redundant Data File Support.....	18
PSMON .....	18
Cluster Support in GSS.....	19
Cluster Components .....	19
Multiple Instance GSS.....	20
Monitoring of Disk Partitions.....	21
Network Deployments and Interfaces .....	22
Deploying the GSS .....	22
Cluster Mode GSS Deployment in GPRS/UMTS Network .....	24
How the GSS Works .....	26
<b>GSS Installation Management .....</b>	<b>27</b>
Installation First Steps .....	28
Step 1 - Verifying System Requirements.....	28
Step 2 - Verifying Hardware Status .....	28
Step 3 - Setting the System Environment Configuration .....	28
Step 4 - Enabling the Database Environment .....	29
Unpacking the Compressed GSS.....	30
Complete GSS .....	32
Installing the Complete GSS - Stand-alone Node.....	32
Using the Installation Script.....	32
Installation Procedure - Stand-alone Node .....	33
Installing the Complete GSS - Node 1 in Cluster .....	39
Using the Installation Script.....	39
Installation Procedure - Node 1 .....	39
Installing the Complete GSS - Node 2 in Cluster .....	46
Uninstalling the Complete GSS - Stand-alone Node.....	54
Uninstalling the Complete GSS - Cluster Nodes.....	55

Upgrading the GSS Stand-alone Node .....	59
Preparing to Upgrade .....	59
Using the Installation Script .....	60
Upgrading a GSS Stand-alone Node .....	60
Upgrading the GSS - Cluster Nodes .....	63
Preparing to Upgrade .....	63
Upgrading Node1 - Primary Node .....	64
Multiple Instances of GSS .....	71
Installing multiple GSS Instances - Stand-alone Node .....	71
Using the Installation Script - Stand-alone Node .....	71
Installation Procedure - Stand-alone Node .....	72
Installing multiple GSS Instances - Node 1 in Cluster .....	78
Using the Installation Script - Cluster Node 1 .....	78
Installation Procedure - Node 1 .....	78
Installing multiple GSS Instances - Node 2 in Cluster .....	86
Uninstalling Multiple GSS Instances - Stand-alone Node .....	92
Using the Uninstallation Script .....	92
Uninstallation Procedure - Stand-alone Node .....	92
Uninstalling multiple GSS Instances - Cluster Nodes .....	97
Using the Uninstallation Script .....	97
Uninstallation Procedure - Cluster Node .....	97
Upgrading multiple GSS Instances - Stand-alone Node .....	103
Preparing to Upgrade .....	103
Using the Installation Script .....	104
Upgrading a GSS Stand-alone Node .....	104
Upgrading multiple GSS instances - Cluster Nodes .....	107
Preparing to Upgrade .....	107
Upgrading Node1 - Primary Node .....	108
Configuring IPMP on GSS Server (Optional) .....	115
Configuring Probe-based IP Multipathing .....	115
Configuring Link-based IP Multipathing .....	116
<b>GTPP Storage Server Administration .....</b>	<b>119</b>
Maintaining the GSS .....	120
Periodic Removal of Processed Data Files .....	120
Using Log Files for Monitoring .....	121
Accessing GSS Commands .....	122
Modifying a GSS Configuration .....	124
GSS Configuration File - Stand-alone Node .....	124
GSS Configuration File - Cluster Nodes .....	137
Troubleshooting the GSS .....	152
Problems During Installation .....	152
Problems Managing a Cluster Node .....	152
Check Status of Node in Cluster .....	152
Change to Stand-alone Mode - Cluster Console Command .....	152
GSS/FileGen/PostgreSQL Not Running .....	153
GSS/FileGen Status .....	153
PostgreSQL Status .....	153
Support Detail Logs for Debugging .....	154
<b>Managing the GSN-GSS Services .....</b>	<b>155</b>
Deployment of GSS with a GSN .....	156
Configuring GGSN Support for GSS .....	158
Configuring SGSN Support for GSS .....	162
Saving the GSN Configuration .....	167





# About this Guide

---

This document pertains to the features and functionality that run on and/or that are related to the Cisco® ASR 5000 Chassis, formerly the Starent Networks ST40.

## Conventions Used

The following tables describe the conventions used throughout this documentation.

Icon	Notice Type	Description
	Information Note	Provides information about important features or instructions.
	Caution	Alerts you of potential damage to a program, device, or system.
	Warning	Alerts you of potential personal injury or fatality. May also alert you of potential electrical hazards.
	Electro-Static Discharge (ESD)	Alerts you to take proper grounding precautions before handling a product.

Typeface Conventions	Description
Text represented as a <i>screen display</i>	This typeface represents displays that appear on your terminal screen, for example: Login:
Text represented as <b>commands</b>	This typeface represents commands that you enter, for example: <b>show ip access-list</b> This document always gives the full form of a command in lowercase letters. Commands are not case sensitive.
Text represented as a <b>command variable</b>	This typeface represents a variable that is part of a command, for example: <b>show card slot_number</b> slot_number is a variable representing the desired chassis slot number.
Text represented as menu or sub-menu names	This typeface represents menus and sub-menus that you access within a software application, for example: Click the <b>File</b> menu, then click <b>New</b>

Command Syntax Conventions	Description
{ <b>keyword</b> or <i>variable</i> }	Required keywords and variables are surrounded by grouped brackets. Required keywords and variables are those components that are required to be entered as part of the command syntax.

Command Syntax Conventions	Description
[ <b>keyword</b> or <i>variable</i> ]	Optional keywords or variables, or those that a user may or may not choose to use, are surrounded by square brackets.
	<p>With some commands there may be a group of variables from which the user chooses one. These are called alternative variables and are documented by separating each variable with a vertical bar (also known as a pipe filter).</p> <p>Pipe filters can be used in conjunction with required or optional keywords or variables. For example:</p> <pre>{ <b>nonce</b>   <b>timestamp</b> }</pre> <p>OR</p> <pre>[ <b>count</b> <i>number_of_packets</i>   <b>size</b> <i>number_of_bytes</i> ]</pre>

## Contacting Customer Support

Use the information in this section to contact customer support.

**For New Customers:** Refer to the support area of <http://www.cisco.com> for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.

**For Existing Customers with support contracts through Starent Networks:** Refer to the support area of <https://support.starentnetworks.com/> for up-to-date product documentation or to submit a service request. A valid username and password is required to this site. Please contact your local sales or service representative for additional information.



**Important:** For warranty and repair information, please be sure to include the Return Material Authorization (RMA) tracking number on the outside of the package.

---

# Chapter 1

## GTPP Storage Server Overview

---

The GTPP Storage Server (GSS) provides an external management solution for the bulk storage of Charging Data Records (CDRs) coming from a GPRS Support Node (GSN) in a GPRS/UMTS network.

The GSS can collect eG-CDRs and/or G-CDRs from a Gateway GPRS Support Node (GGSN) or the GSS can collect any of the following CDR types from a Serving GPRS Support Node (SGSN):

- M-CDRs
- S-CDRs
- SM-MO-CDRs
- SM-MT-CDRs

This overview provides general information about the GSS including:

- [Product Description](#)
- [System Requirements and Recommendations](#)
- [IP Multipathing \(IPMP\) on GSS Server \(Optional\)](#)
- [Features of the GSS](#)
- [Network Deployments and Interfaces](#)
- [How the GSS Works](#)

## Product Description

The GSS enhances the mobile carrier's ability to manage the CDRs. Running on standard carrier-grade servers in either a stand-alone or cluster-aware deployment, there are no practical limits on the period for storage thus ensuring high availability.

The GSS provides redundant/backup CDR storage for the billing/charging data by enabling the GGSN to simultaneously send CDRs to both the GSS and the Charging Gateway Function (CGF).

The GSS FileGen utility generates proprietarily encoded CDR files for transfer via FTP or SFTP to offline Billing System (BS).

The GTPP storage server comprises the following feature components:

- GSS server application software
- PostgreSQL database
- FileGen utility
- Process monitor utility (PSMON)
- Cluster mode support

## Partnering with a GSN

The GSS is an “external application” product that resides on a server separate from the ASR 5000 GSN. GSS is only accessible if you have purchased this product separately and purchased and installed a GSS feature license on your ASR 5000 GSN system.


Prior to attempting to connect the GSS to the GSN, it is recommended that you:

- Step 1** Select the stand-alone or cluster mode configuration that best meets your service model (check the [System Requirements and Recommendations](#) section in this chapter).
- Step 2** Configure the required server elements as described in the vendor's documentation.
- Step 3** Install and configure the GSS application (see the *GSS Installation Management* chapter in this guide).
- Step 4** Setup the GSS support on the GSN (see the *Managing the GSN-GSS Services* chapter in this guide).

## System Requirements and Recommendations

This section identifies the minimum system requirements for the GTPP Storage Server. This section also describes any specific software requirement for a particular application installation.

---


 **Important:** The hardware required for these components may vary depending on the number of clients that require access, other components managed, and other variables.

---

### Minimum System Requirements for Stand-alone Deployment

- Sun Microsystems Netra™ T5220 server
  - 1 x 1.2GHz 8 core UltraSPARC T2 processor with 16GB RAM
  - 2 x 146GB SAS hard drives
  - Internal CD-ROM drive
  - AC or DC power supplies depending on your application
  - Quad Gigabit Ethernet interfaces (10/100/1000 Gigabit Ethernet)


---

 **Important:** It is recommended that you have separate interfaces (in IPMP) for mediation device and chassis. Also, for given IPMP, the two interfaces should be on different cards.

---

- Operating Environment:
  - Solaris 9 installed using the End User System support 64-bit software group with the latest available patches from Sun Microsystems.
  - Solaris 10 with Patch number 137137-09 dated on or after July 16, 2007 to Nov 2008.
- PCI-based video card or Keyboard-Video-Mouse (KVM) card (optional)

---

 **Important:** If you plan to install software and maintain the servers and applications remotely, it is recommended that you use an X-Windows client.

---

### Minimum System Requirements for Cluster Deployment

Hardware and software requirement mentioned in this section is for single node in cluster. For additional node additional number of hardware and software are required.

- Sun Microsystems Netra™ T5220 server
  - 1 x 1.2GHz 8 core UltraSPARC T2 processor with 16GB RAM
  - 2 x 146GB SAS hard drives
  - Internal CD-ROM drive
  - Quad Gigabit Ethernet interfaces (10/100/1000 Gigabit Ethernet)



**Important:** It is recommended that you have separate interfaces (in IPMP) for mediation device and chassis. Also, for given IPMP, the two interfaces should be on different cards.

- PCI-based video card or Keyboard-Video-Mouse (KVM) card (optional)
- Fiber Channel (FC) based Common Storage System for Servers (Sun Storage Tek 2540)
- Two 4GB dual port PCI FC HBAs
- Dual RAID Controllers
- 5 x 300GB 15K drives
- AC or DC power supplies depending upon your application
- Optical 5 meter null ethernet cable
- Operating Environment:
  - Solaris 9 installed using the End User System support 64-bit software group with the latest available patches from Sun Microsystems.
  - Solaris 10 with Recommended Patch Cluster dated on or after July 16, 2007 to Nov 2008.
- Sun Cluster Software version 3.2 or later installed on node.



**Important:** If you plan to install software and maintain the servers and applications remotely, it is recommended that you use an X-Windows client.

## Default Ports for GSS

The various components of the GTPP storage server use specific TCP/UDP ports by default. The following table lists the default ports.

*Table 1. Default TCP/UDP Port Utilization*

Port Number	Usage
TCP/UDP	
5432	Used by the PostgreSQL database server with the GSS.

Port Number	Usage
50000	Used by the GSS Server for communication with the GSN.
50001	Used by the GSS FileGen with the GSS.
32838 to 32862	Used by the Postgres Client.
22	This is the SSH port used by mediation system to access generated data files for further processing.
21	This is the FTP port used access generated data files for further processing.
<b>Ports used in cluster mode</b>	
9444	This is the CRNP server port used by solaris system to gather system resource information.
9900	This is the CRNP client port used by GSS to receive system resource information from CRNP server to generate alarms if any.

## GSS Hardware Sizing and Provisioning Guidelines

In addition to the minimum system requirements indicated in the [Minimum System Requirements for Stand-alone Deployment](#) and [Minimum System Requirements for Cluster Deployment](#) sections, the following section offers information that can help you to plan hardware sizing needs, based on the exact deployment scenario that you are using.

### Hard Drive Partition Recommendations

Following is the partition scheme required for GSS application:

- Root partition (/) should be at least 15 gigabyte (GB).
- The swap partitions (/tmp, /var/run) should be at least 3 GB.
- /globaldevices should be at least 1 GB - This is applicable for Cluster mode only.
- /opt should be at least 10 GB
- /export/home should be the partition used for GSS and PostgreSQL.
  - In Stand-alone mode this partition should have at least 20 GB free disk space to allow for longer-term storage of the CDR files and other archived databases.
  - In case of Cluster mode installation PostgreSQL and CDR storage will be on /sharedisk for all cluster node hence may not require 20 GB free disk space.

A typical CDR can be 200 Bytes in size. Based on this, the approximate file size with 4 Million CDRs per hour and backup for 2 days, the formula used to calculate the amount of space needed to backup this information is:

$$200 \times (\#\_of\_CDR\_per\_hour) \times 48 \times 1.5 = \text{Backup space on Hard disk in Bytes.}$$

## IP Multipathing (IPMP) on GSS Server (Optional)

IPMP or IP multipathing is a facility provided by Solaris® to provide physical interface failure detection and transparent network access failover for a system with multiple interfaces on the same IP link. IPMP also provides load spreading of packets for systems with multiple interfaces.

For IPMP configuration, refer to the *Configuring IPMP on GSS Server* section in the *GSS Installation Management* chapter.



**Important:** IPMP is a feature supported on Sun® Solaris® provided by Sun Microsystems. The configuration is included in the *System Administration Guide*. For more information, refer to the *Sun documentation*.

---

# Features of the GSS

This section describes the various features of GSS application.

## GSS Server Application

This software application receives the CDRs from the GSN and stores them in database tables. It also provides a mechanism to send ACK responses to the GSN.

## PostgreSQL Database Engine 8.4.4

The GSS application uses this database engine to process and store the information received from the GSN and the records generated by the GSS application. It is required that the PostgreSQL database engine resides on the same server as the GSS application.

## GSS FileGen Utility


The GTPP Storage Server has a file generation utility called the GSS FileGen. It is used to generate the CDR files for the billing systems which do not have direct billing interface with the GSN.

The GSS FileGen saves the CDRs stored in the GSS database to the disk files.

## File Format Encoding for CDRs

The file format determines the information organization and structure -- format -- of the generated data files. All file formats are different and are customizable.

---

 **Important:** If none of the following formats meet your needs, you should contact your support representative to enquire about obtaining a customized file format.

---

The GSS FileGen utility supports the following file formats for CDRs:

- **starent Format:** This default file format encodes CDRs according to the following conventions:
  - **Header:** No header
  - **Contents:** *CDR1CDR2CDR3...CDRn*
  - **EoF marker:** \n
  - **File name format:**

GSN\_<date>+<time>\_<total-cdrs>\_file<fileseqnum>

GSN\_<date>+<time>\_<total-cdrs>\_unacked\_file<fileseqnum>

- **custom1 Format:** This file format encodes CDRs according to the **starent** file format explained above.



**Important:** The use of either **starent** or **custom1** file formats, imposes a few specific reactions: - files are generated without an extension; acknowledged and unacknowledged files are differentiated by their file names; the system deletes all the files after reaching the maximum storage period (1-7 days) configured during GSS configuration.

- **custom2 Format:** This customer-specific file format encodes CDRs according to the following conventions:
  - **Header:** 24 byte header incorporating the following information:

Field	Description	Value
0x00 - 0x03	Offset	Offset from EoH to first Unread CDR (4 Bytes)
0x04 - 0x07	Encoding	Basic Encoding Rule (BER) i.e. 1 (4 Bytes)
0x08 - 0x0b	Number of CDRs	Total number of CDRs in the file (4 Bytes)
0x0c - 0x0f	Number of read CDRs	Total number of read CDRs in the file (4 Bytes)
0x10 - 0x13	File size	Size of CDR file in bytes (4 Bytes)
0x14 - 0x17	Abstract Syntax Notation One (ASN.1) format definition version	ASN.1 definition version information (4 Bytes)

- **Contents:** *LEN1CDR1LEN2CDR2LEN3CDR3...LENnCDRn*

- **EoF marker:** No EoF marker

- **File name format:**

GSN\_<date>+<time>\_<total-cdrs>\_file<fileseqnum>.u

- **custom3 Format:** This customer-specific file format encodes CDRs according to the following conventions:

- **Header:** No header

- **Contents:** *CDR1CDR2CDR3...CDRn*

- **EoF marker:** No EoF marker

- **File name format:**

GSN\_<date>+<time>\_<total-cdrs>\_file<fileseqnum>.u




**Important:** The use of either **custom2** or **custom3** file formats imposes the following actions: - files are generated with the **.u** file extension (indicating an unprocessed file to the billing system); - the GSS system deletes files with **.p** extension as part of periodic clean-up.

- **custom4 Format:** This custom4 format was created to support writing CDRs in blocks. This file format is similar to custom3 file format except CDRs will be written in 2Kbyte blocks in a file.
  - **Header:** No Header
  - **Contents:** CDR1|CDR2FFFFFF|CDR3FFFFFF[..CDRnFFFF]
    - where | represents the end of a 2k block
  - **EoF marker:** No EoF marker
  - **File name format:**

```
<GSN_Location>_<date>+<time>_<total-cdrs>_file<fileseqnum>.u
```

---


 **Important:** With file format **custom4**, the files are generated with **.u** file extension indicating an unprocessed file by the billing system. Typically, the billing system would rename the file with **.p** extension after processing the files with CDR information. This also informs the GSS system that the file can be deleted during periodic cleanup.

---

- **custom5 Format:** This file format is similar to custom3 file format except that the sequence number for CDR file name is of six digits in length ranging from 000001 to 999999.
  - **Header:** No Header
  - **Contents:** CDR1CDR2CDR3...CDRn
  - **EoF marker:** No EoF marker
  - **File name format:**

```
<GSN_Location>_<date>+<time>_<total-cdrs>_file<fixed-length-seqnum>.u
```

---

 **Important:** This release of GSS does not support custom6 file format.

---

- **custom7 Format:** This customer-specific file format contains CDRs converted from ASN.1 format to ASCII format according to the following conventions. Each line in the file consists of one CDR which contains 33 parameters occupying 491 bytes.
  - **Header:** No Header
  - **Contents:** CDR1CDR2CDR3...CDRn
  - **EoF marker:** No EoF marker
  - **File name format:**

```
Processed_02_YYYYMMDDhhmmss.cdr
```
- **custom8 Format:** This customer-specific file format encodes CDRs according to the following conventions:
  - **Header:** No Header
  - **Contents:** CDR1CDR2CDR3...CDRn
  - **EoF marker:** No EoF marker
  - **File name format:**

`<node-id-suffix>_<date>_<time>_<fixed-length-seq-num>.u`



**Important:** The custom2 to custom8 file formats are customer-specific. For more information on the file formats, contact your local sales representative.

For more information on CDR accounting attribute elements, refer to the *AAA and GTPP Interface Administration and Reference Guide*.

## Redundant Data File Support

The FileGen utility includes an additional feature to generate redundant GSS files. When this feature is enabled, the FileGen utility automatically creates a directory called `/<GSS_install_dir>/data_redundant` (name cannot be changed). After the original data file is created and stored in the `/<GSS_install_dir>/data` directory, the FileGen utility creates a hard link between the `/<GSS_install_dir>/data_redundant` directory and the same tmp file that was used to create the original data file. Effectively, this creates a copy and stores a hard link duplicate in this redundant directory.

The redundant directory is in the same partition and cannot be moved. Hardlinked means that the redundant files are not deleted if/when the original files are deleted.

By default, this feature is disabled. It can be enabled during the installation of the GSS application (see the installation procedure later in this guide) or it can be enabled/disabled at anytime by using a text editor to modify the appropriate lines in the GSS configuration file (`gss.cfg`):

```
#Key: Enable_Redundant_File

#Flag to indicate whether to enable redundant file creation in path parallel to
#primary data path. For example <gss_dir>/data_redundant

#Value : yes/no

#Default : no

Enable_Redundant_File = y
```

## PSMON

The PSMON is a UNIX process monitor utility that starts when GSS starts and then runs in the background as a fully functional background daemon, capable of logging to syslog and log file with customizable E-mail notification facilities.

PSMON monitors the PostgreSQL Database, GSS, and FileGen processes. The PSMON scans the operating system process table and, using the set of rules defined in the configuration file, respawns any dead processes.

## Cluster Support in GSS

The cluster mode feature enables GSS to provide high availability and critical redundancy support to retrieve CDRs in failure of any one of the system. A GSS cluster is two or more GSS systems, or nodes, that work together as a single, continuously available system to provide applications, system resources, and data to GSS users. Each GSS node on a cluster is a fully functional, stand-alone system. However, in a clustered environment, the GSS nodes are connected by an interconnected network and work together as a single entity to provide increased availability and performance.

Highly available clusters provide nearly continuous access to data and applications by keeping the cluster running through failures that would normally bring down a single Server system.

A cluster offers several advantages over traditional single-server systems. These advantages include:

- Support for failover and scalable services.
- Capacity for modular growth.
- Low entry price compared to traditional hardware fault-tolerant systems.
- Reduce or eliminate system downtime because of software or hardware failure.
- Ensure availability of data and applications to GSS user, regardless of the kind of failure that would normally take down a single-server system.
- Provide enhanced availability of the system by enabling you to perform maintenance without shutting down the entire cluster.

## Cluster Components

Following are the cluster components work with GSS to provide this functionality:

- **GSS Cluster Node**

A GSS cluster node is a GSS server that runs both the GSS Application software and Cluster Agent software. The Cluster Agent enables carrier to network two GSS nodes in a cluster. Every GSS node in the cluster is aware when another GSS node joins or leaves the cluster. Also, every GSS node in the cluster is aware of the resources that are running locally as well as the resources that are running on the other GSS cluster nodes.


Each GSS cluster node is a stand-alone server that runs its own processes. These processes communicate with one another to form what looks like (to a network client) a single system that cooperatively provides applications, system resources, and data to GSS users.

- **Common Storage System**

A common storage system is a Fiber Channel (FC) -based cluster storage with FC drives for the servers in the cluster environment. It is interconnected with GSS cluster nodes with carrier class network connectivity to provide high level redundant storage and backup support for CDRs. It serves as common storage for all connected GSS cluster nodes.

This system provides high storage scalability and redundancy with RAID support.

---

 **Important:** For information on Switching CDRs from HDD to GSS and Switching CDRs from GSS to HDD procedures, refer to the *AAA and GTPP Interface Administration and Reference Guide*.

---

## Multiple Instance GSS

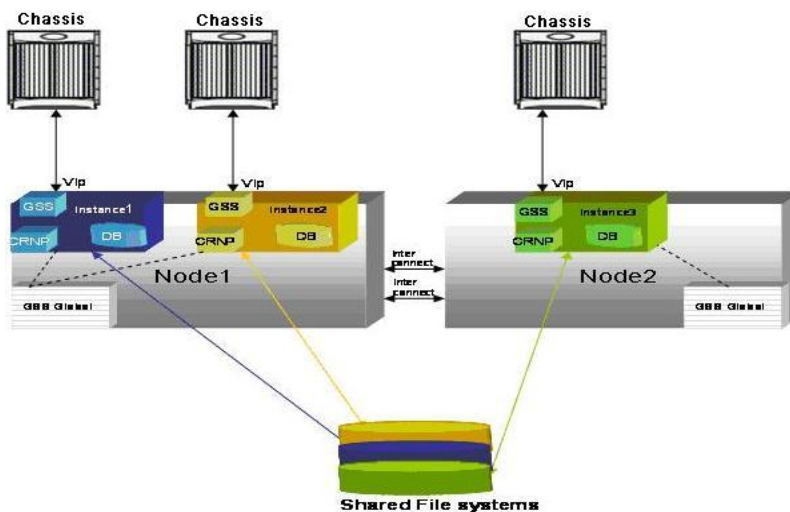
This feature enables support for multiple data streams from one server or a single cluster setup to utilize multiple instances of GSS with a single installation and multiple databases. In a cluster setup, there is only one installation per node. During installation, GSS is installed at a fixed location (`/opt/gss_global` directory). The initial GSS installation does not create any GSS instance. Once GSS is installed on both the nodes, the `/opt/gss_global/make_gss_instance` script utility creates instances as and when needed and validates the conflicting ports/username across the instances.

For all instances on the node, only one set of binaries and scripts are used. Each instance has its own configuration file, log directory, tools directory and separate PostgreSQL database. The alarms and events generated by each instance are sent to its corresponding chassis. Individual GSS instance can also be stopped, started or switched over. Upgrade is smooth and involves minimum down time as possible.

Each GSS instance can be uninstalled separately and will not have any impact on the other instances. Global installation can be only uninstalled if there are no instances configured or running on the system.

The following figure explains the architecture of multiple GSS instances in a cluster setup.

**Figure 1. Multiple Instances GSS**



The advantages of this feature include:

- Only one installation required for multiple instances
- One binary used across all the instances on the node
- Upgrading one set of binaries upgrades all the instances
- In cluster mode resource groups, instances can be balanced across the nodes

For more information on the installation, uninstallation and upgrade procedures for multiple GSS instances, refer to *Multiple Instances of GSS* section in the *GSS Installation Management* chapter.

## Monitoring of Disk Partitions

This feature enables support for disk monitoring of shared postgres and gss installation disk partition along with GSS data files disk partition. This feature enables sending an alarm or a notification based on the available disk space for postgres database and GSS base directory. This feature is supported only for single instance GSS, and for GSS in cluster mode.

This feature can be enabled after installation by configuring *Notif\_Disk\_Usage\_Postgres\_Database* and *Notif\_Disk\_Usage\_Gss\_Base* parameters from gss configuration file and there is no configuration support from installation script or during installation. For information on configuring these parameters, refer to *Modifying a GSS Configuration* section in the *GTPP Storage Server Administration* chapter of this guide.



**Important:** This feature does not support backward compatibility and hence GSN build should always match with GSS build. If GSN build and GSS build mismatches, then disk usage alarm and GSN Storage Server Status CLI will not work as expected at GSN side and some malfunction may occur. In this case GSN and GSS will be functional only if disk usage alarm is disabled and Storage Server Status CLI is not used.

---

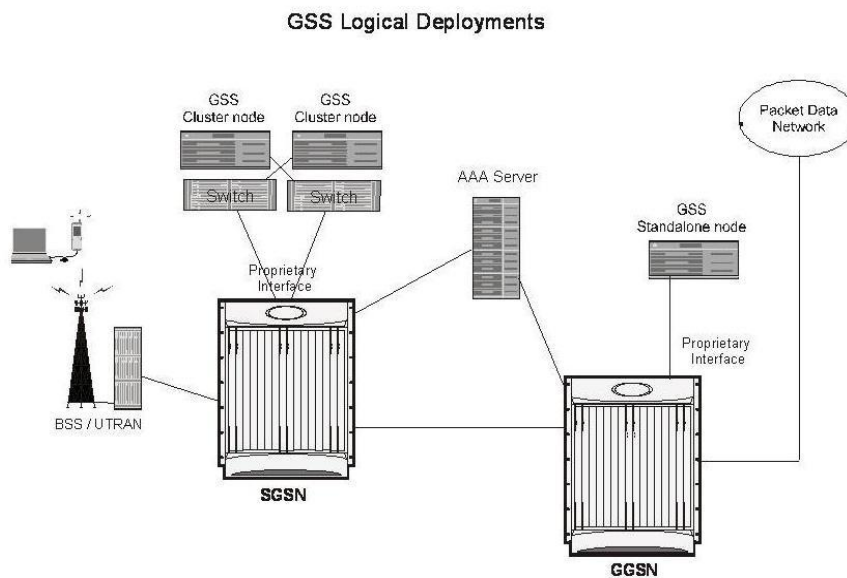
## Network Deployments and Interfaces

The GSS, in either a stand-alone or a cluster configuration, partners with a GSN (either an SGSN or a GGSN) in a GPRS/UMTS network to support a secure accounting solution. Optionally, other elements are included as needed such as a billing/mediation system, a RADIUS AAA server, a fiber channel common storage server, and/or a Charging Gateway Function (CGF).

## Deploying the GSS

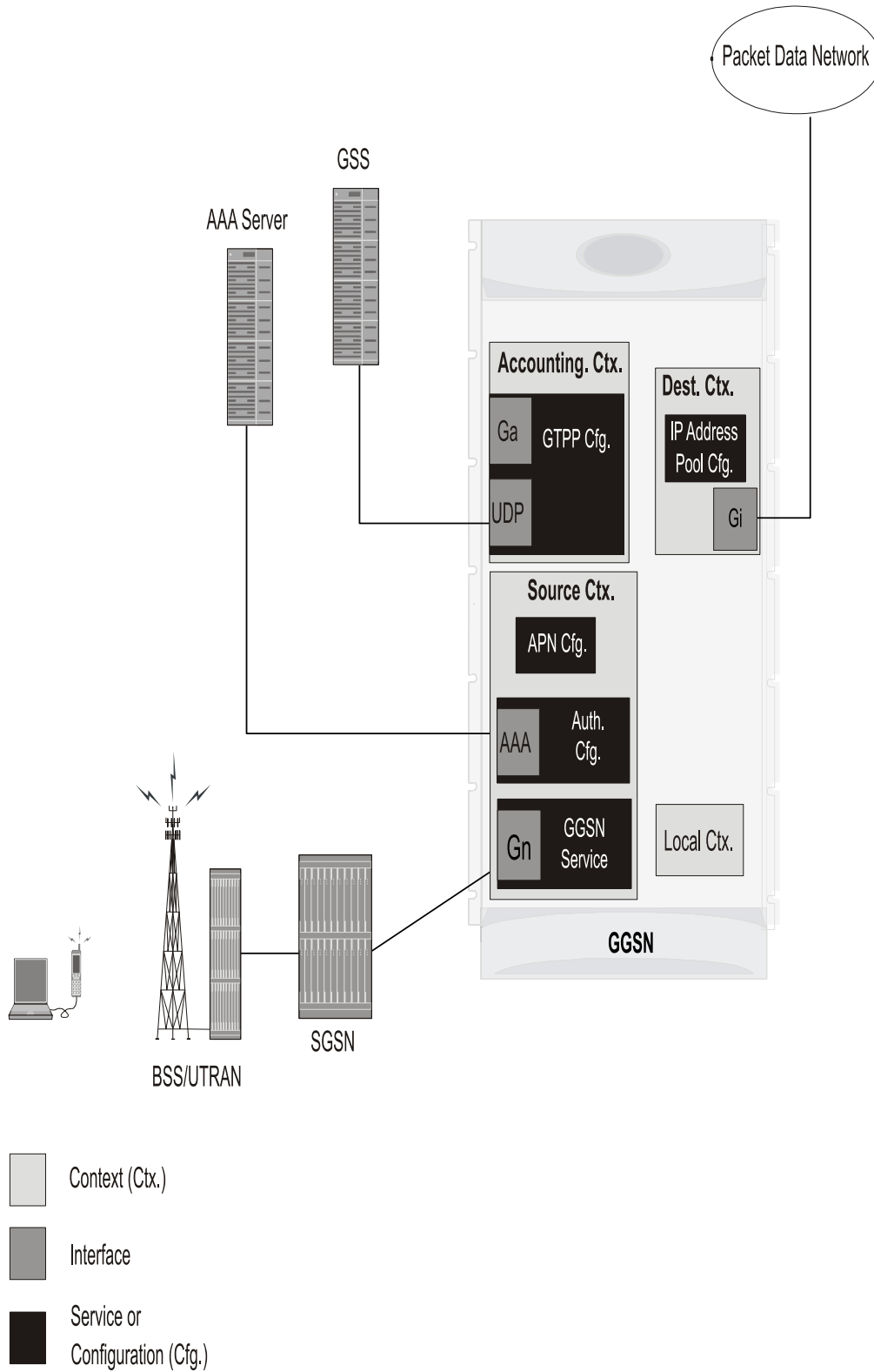
The following figure shows two typical deployments of the GSS in a GPRS/UMTS network.

**Figure 2. GSS in GPRS/UMTS Network**



The SGSN (SGSN Service) and the GGSN (GGSN Service) incorporate a range of user-defined and default contexts for the accounting functions - as illustrated in the following figure.

Figure 3. GGSN Contexts and Interfaces



The logical accounting context in the SGSN Service on an SGSN and the GGSN Service on a GGSN facilitate:

- GPRS Tunneling Protocol Prime (GTPP) configuration
- UDP interface to the GSS
- Optional Ga interface to a Charging Gateway Function (CGF)
- Optional Network-requested PDP context processing

The source context of the GSN usually includes the

- Access Point Name (APN) configuration
- RADIUS authentication configuration (Auth.cfg) and the interface (AAA) to the authentication server
- GGSN or SGSN service(s) and Gn interface to another GSN

The GGSN destination context (not supported by SGSN) facilitates:

- IP address pools
- Gi interface to the Packet Data Network (PDN)

In order to support a GSS, the GSN system is configured with two components:

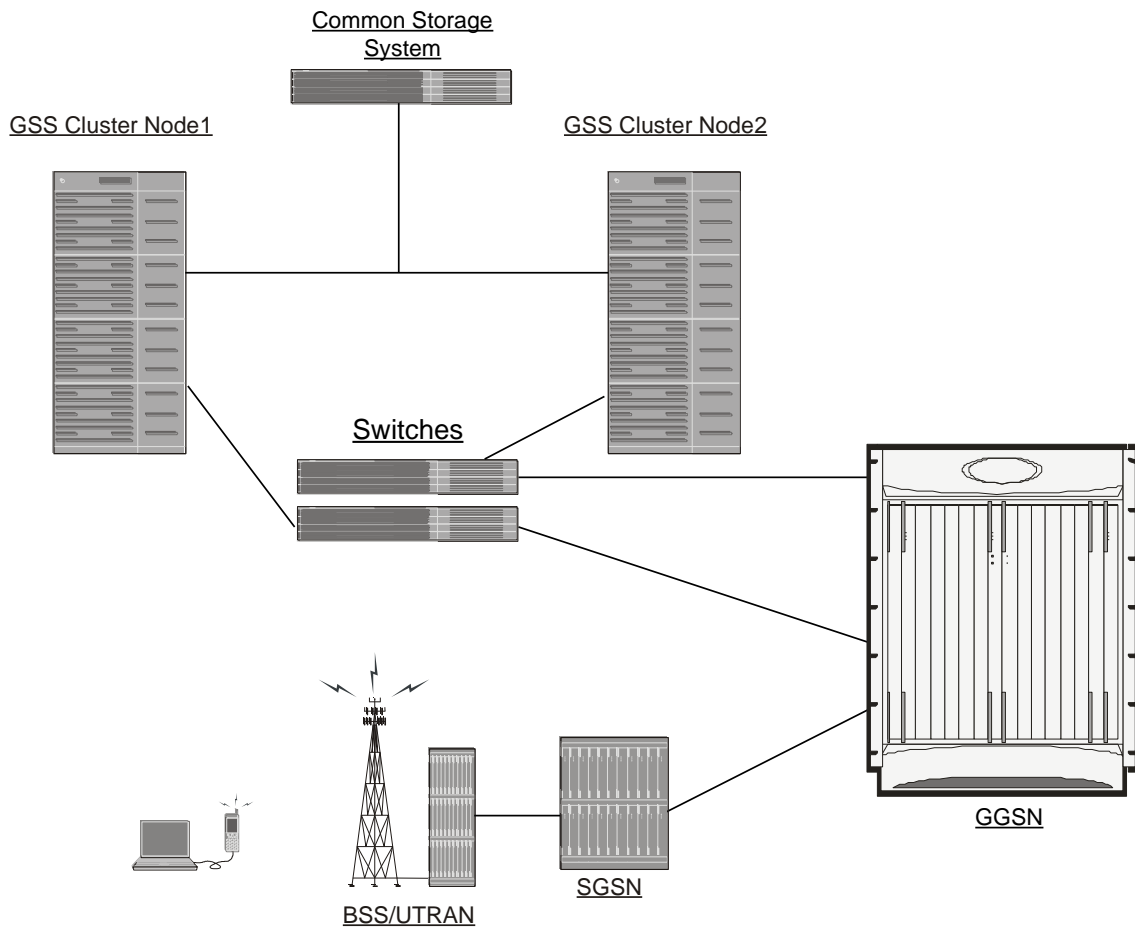
- **GTPP Storage Server (GSS)** is configured in the same context as the GSN service(s) or any other accounting context. The configuration of the GSN initiates the tasks that communicate with the GSS.
- **UDP interface** on the GSN is bound to the GTPP Storage Server (GSS). The UDP interface is a proprietary interface used by the GSN system to communicate with the GSS.

## Cluster Mode GSS Deployment in GPRS/UMTS Network

The following figure shows a typical deployment of the cluster-aware GSS nodes in a GPRS/UMTS network with a Common Storage System. The GSS nodes, connecting through switches, could be connected to either a GGSN or an SGSN. As described earlier, the cluster nodes connect to the GGSN source context or the SGSN accounting context via the UDP interface.

The GSS cluster nodes process as stand-alone nodes with one in primary or active mode and the other in standby mode as a redundant backup system.

Figure 4. GSS Cluster Nodes in a GPRS/UMTS Network



## How the GSS Works

The GSS and the GSS FileGen utility need to be configured to archive incoming records and export them to CDR files. The GSS generates the CDR files with a customer specific format. These generated CDR files can then be pulled (via FTP or SFTP) and used by the carrier's billing system.

The following describes how the GSS interoperates with a GSN:

1. Once the CDRs are generated, the GSN creates a transid (transaction ID, a unique 4 Byte running counter between GSN and GSS), and sends the set of the generated CDRs along with this transid and a STORE request to the GSS through AAA Proxy (GSN) on a proprietary interface (based on UDP).
2. On receipt of the set of CDRs and the transid, the GSS stores them in the Postgres database and sends ACK response to the GSN.
3. The GSS FileGen utility retrieves records from the database and generates CDR files. As explained in [File Format Encoding for CDRs](#) section, these CDR files have vendor specific extensions and formatting for the billing system to use.

To generate a CDR file, the FileGen utility performs the following tasks:

- It starts writing a raw file in `/<GSS_install_dir>/data` directory with name tmp.
- Based on the CDR counts per file or the file life expiry, it saves the target file with .u extension using the specified file naming format.

Once the files are generated, then the files with .u extensions in the `/<GSS_install_dir>/data` directory can be pulled by a billing system for the processing of the charging details.

Depending upon the billing system, after processing the files pulled by the billing system can be stored with .p extension. The processed files with .p extensions can then be removed by the clean-up script based on the Maximum Storage Period for generated/processed data files.

4. All records written to the CDR file are deleted immediately from the database, without consideration of the configured archive period on the GSS.
5. If CDRs are not written to CDR files using the GSS FileGen, then all CDRs in the database are kept for a pre-defined period of time (typically not more than 7 days). After the period expires, the GSS Clean-up utility (`cleanup.sh`) deletes them.

# Chapter 2

## GSS Installation Management


---

This chapter provides information and procedures to perform installations and removals (uninstallations) of the GTPP Storage Server (GSS) software application with all of its various components.

This chapter also includes procedures for upgrading the GSS software application and the PostgreSQL database.

- [Installation First Steps](#)
- [Unpacking the Compressed GSS](#)
- [Complete GSS](#)
- [Multiple Instances of GSS](#)
- [Configuring IPMP on GSS Server \(Optional\)](#)

---

 **Important:** It is recommended that you select the deployment and configurations that best match your service requirements. All elements must be setup prior to attempting any of the procedures detailed in this chapter. To perform any of the procedures listed in this chapter, you must be logged into the server as a **root** user.

---

# Installation First Steps

The following procedure is relevant for both stand-alone and cluster nodes.

Before you begin the installation process, there are four steps you should take to ensure a quick and successful installation of the GSS. Following completion of these steps, you will need to unpack the compressed GSS application components.

## Step 1 - Verifying System Requirements

This section lists the basic requirements needed for the system verification.

- Verify the correct operating system. Enter the UNIX command:  
`uname -a`
- Refer to the *System Requirements and Recommendations* section in the *Overview* chapter of this guide to confirm that your system meets the minimum requirements for:
  - stand-alone deployment
  - cluster mode deployment

## Step 2 - Verifying Hardware Status

The first thing that you need to do is to ensure that the system hardware has been provisioned properly for your application. This includes:

- Verify that the hard disk has been partitioned properly. Enter the UNIX command:  
`df -kh`  
Refer to the hard disk partitioning recommendations outlined in the *GSS Hardware Sizing and Provisioning Guidelines* section in the *Overview* chapter of this guide.
- Verify network connectivity between the GSS node and the SGSN or the GGSN chassis through the management LAN.

## Step 3 - Setting the System Environment Configuration

This step is required to define how the PostgreSQL database engine processes, stores, and retrieves information contained in the various databases stored using the UNIX file subsystem.

Failure to configure these settings may cause data loss and will minimally cause errors in the operation.

Use a text editor to add the following values to the bottom of the system file in the */etc* directory and then complete *Step 4* before beginning the installation of the GSS application components.

```
set msgsys:msginfo_msgmnb=65536
set msgsys:msginfo_msgtql=1024
set shmsys:shminfo_shmmax=33554432
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=256
set shmsys:shminfo_shmseg=256
set semsys:seminfo_semmap=256
set semsys:seminfo_semmni=512
set semsys:seminfo_semmns=512
set semsys:seminfo_semmsl=32
```

## Step 4 - Enabling the Database Environment

After adding the above values to the system file in the */etc* directory, restart the system before installation of the GSS application and components. Enter the UNIX command:


**reboot**

Once you have completed the installation preparation, then you are ready to unpack the compressed GSS software files. This process is explained in the next section.

## Unpacking the Compressed GSS

The components that comprise the GSS application software are bundled and distributed in a single compressed file package. Your sales representative will inform you how to download the appropriate GSS package for your requirements.

---

 **Important:** In the GSS Software Releases prior to 9.0.101 build, the installation file is distributed with a **.tar.gz** extension. In the GSS Software Release 9.0.101 and later, this file is distributed in **.zip** format.


---

**Step 1** Login to the GSS server.

**Step 2** Create a directory, at the root level, into which you can download the compressed GSS application package (and possibly future versions) - for example:

```
mkdir /packages
```

---

 **Important:** Remember that within a procedure, information inside angle brackets *<variable>* represents a variable that can be defined by either the user or generated by the system. For example: Create the */<package>* directory to hold the application packages.

---


**Step 3** Copy or move the compressed GSS application file to the */<packages>* directory.

**Step 4** Unzip the compressed file by entering the following command:

```
unzip gss_x.x.xx_solaris_sparc.zip
```

*x.x.xx* is the version number of the GSS software distributed in the zip file. For example, *gss\_9.0.101\_solaris\_sparc.zip*.

---

 **Important:** To unzip the **.gz** package file, use **tar -xvf <file\_name>** command

---

```
drwxrwxr-x 2 1071 100 512 Dec 12 15:01 .
drwxr-xr-x 3 root other 512 Dec 15 09:54 ..
-r-xr-xr-x 1 1071 100 1889 Dec 12 15:01 Global_Gss_Uninstall.sh
-r--r--r-- 1 1071 100 1094 Dec 12 15:01 README_INSTALL
-r--r--r-- 1 1071 100 4459 Dec 12 15:01 README_UNINSTALL
-rwxr-xr-x 1 1071 100 118784 Dec 12 15:01 StarentGss.tar
-r-xr-xr-x 1 1071 100 4381 Dec 12 15:01 cluster_db_upgrade
-r-xr-xr-x 1 1071 100 103111 Dec 12 15:01 create_gss_instance
-rwxrwxr-x 1 1071 100 13414400 Dec 12 15:00 db.tar
-r--r--r-- 1 1071 100 1168 Dec 12 15:01 gss.env
```

```
-rwxrwxr-x 1 1071 100 96828928 Dec 12 15:01 gss.tar
-r--r--r-- 1 1071 100 6459 Dec 12 15:01 gss_db.sql
-r--r--r-- 1 1071 100 1927 Dec 12 15:01 gssclusterconfig
-r-xr-xr-x 1 1071 100 19751 Dec 12 15:01 inst
-r-xr-xr-x 1 1071 100 3224 Dec 12 15:01 inst_db
-r-xr-xr-x 1 1071 100 75779 Dec 12 15:01 inst_serv
-r-xr-xr-x 1 1071 100 2835 Dec 12 15:01 make_gss_instance
-r-xr-xr-x 1 1071 100 4463 Dec 12 15:01 make_postgres_instance.sh
-r--r--r-- 1 1071 100 1328 Dec 12 15:01 nvpair.dtd
-r-xr-xr-x 1 1071 100 1513 Dec 12 15:01 postgresctl
-r--r--r-- 1 1071 100 1012 Dec 12 15:01 sc_event.dtd
-r--r--r-- 1 1071 100 1067 Dec 12 15:01 sc_reply.dtd
```


**Step 5** After the installation files are unpacked, proceed to the procedures for your node type: either [Installing the Complete GSS - Stand-alone Node](#) or [Installing the Complete GSS - Node 1 in Cluster](#) .

To install multiple instances of GSS, refer to [Multiple Instances of GSS](#) section.

## Complete GSS

This section includes procedures for installation, uninstallation and upgrade to support multiple instances of GSS on a same cluster setup and standalone mode.

---

 **Important:** If you are deploying multiple instances of GSS, please refer to the procedures outlined in [Multiple Instances of GSS](#) section.

---

- [Installing the Complete GSS - Stand-alone Node](#)
- [Installing the Complete GSS - Node 1 in Cluster](#)
- [Installing the Complete GSS - Node 2 in Cluster](#)
- [Uninstalling the Complete GSS - Stand-alone Node](#)
- [Uninstalling the Complete GSS - Cluster Nodes](#)
- [Upgrading the GSS Stand-alone Node](#)
- [Upgrading the GSS - Cluster Nodes](#)

## Installing the Complete GSS - Stand-alone Node

This section describes the process for installing the GSS server application, and all of the associated GSS components, for a stand-alone deployment.

### Using the Installation Script


Installation is accomplished using the **inst\_serv** script. It provides a menu-driven interface with question prompts. Most prompts display default values or information derived from the server's current setup - such as IP addresses for configured interfaces.

The following information will help you use the installation script most effectively:

- **Ctrl-C** will abort the installation process at any time during the procedure.
- The script senses whether the server is a stand-alone node or a cluster node.
- To accept defaults and continue the process, enter **n** as your choice for any prompt.
- To change a displayed value, enter the menu item number to display an entry field or a list of possible values.

The information from the prompts is used to generate the GSS configuration file (**gss.cfg**). This file can be changed at anytime after the installation.

---

 **Important:** It is recommended that you fill in path prompts only after you have created the directories to be used.

---

## Installation Procedure - Stand-alone Node

The following procedure assumes that you are logged in to the GSS server with root privileges and that you are starting at the root directory level.

**Step 1** Change to the `/<packages>/gss_<version>` directory where you stored the GSS application software in *step 5* of the previous section.

**Step 2** Locate the installation script file `inst_serv` and execute the following command:

```
./inst_serv
```



**Important:** This script will check the version of the operating system installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Stand-alone Deployment* section, the script will abort the GSS installation.

The following appears, with pauses for validation, after entering the `inst_serv` command:

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
required changes in '/etc/system' file, restart the machine to get these
changes reflected and then start installation again.
```

```
set msgsys:msginfo_msgmnb=65536
```

```
set msgsys:msginfo_msgtql=1024
```

```
set shmsys:shminfo_shmmax=33554432
```

```
set shmsys:shminfo_shmmin=1
```

```
set shmsys:shminfo_shmmni=256
```

```
set shmsys:shminfo_shmseg=256
```

```
set semsys:seminfo_semmmap=256
```

```
set semsys:seminfo_semmni=512
```

```
set semsys:seminfo_semmns=512
```

```
set semsys:seminfo_semmsl=32
```

```
Standalone Mode Installation [in standalone environment]...
```

**Step 3** If you did not make the changes to the system file, then abort now (**CTRL-C**), make the changes to the system file, and then reboot. After rebooting, begin the installation procedure again.

If you made changes to the system file as in the [Installation First Steps](#) section, then continue to the next step.

```
GTPP Storage Server installation directory
```

```
Path where Gss will be installed :/opt
```

```
GSS Installation dir [/opt] ? </home/export/install_8_0_xx>
```

**Step 4** Enter the name of the directory where the GSS active components are to be installed, for example `<install_dir>`. It is recommended that you include this directory at the root level. The installation script creates the directory if needed.

Shortly after typing `<install_dir>` and pressing **Enter**, the following appears:

```
Entering n will save configuration values and take you to next
configuration, To change the default values, enter option number
```

```
*** PostgreSQL installation configuration ***
```

```
1) PostgreSQL port : 5432
```

```
2) PostgreSQL login : postgres
```

```
3) PostgreSQL passwd : postgres
```

```
n) Proceed to next configuration
```

```
a) Abort Installation
```

```
Enter Your Choice : [n] ?
```

```
Entering n will save configuration values and take you to next
configuration, To change the default values, enter option number
```



**Important:** All values that appear initially for this menu are system defaults and you do not need to make changes if the values are acceptable.

---

**Step 5** Enter the line number to change a parameter value, if needed. Then enter **n** to save changes (if made) or defaults and move to the next menu.

```
*** GSS Configuration Parameters ***
```

```
1) File Format for data files : starent
```

```
2) Hard Limit Interval for File Generation (mins) : 0
```

```
3) Support for LRSN rewrite : n
```

- 4) Encoding of IP Address in binary format : n
- 5) Enable redundant data file support : n
- 6) GSN Location : GSN
- p) Go back to previous menu
- n) Proceed to next configuration
- a) Abort Installation



**Important:** The GSS and FileGen are set to run in archive mode by default. The archive mode, used in deployments that do not include CGFs, instructs the server to save records to a file.

**Step 6** Enter a line number to change a parameter value, if needed. This can be repeated as often as necessary to change parameter values. Then enter **n** to save changes (if made) or defaults and move to the next menu.

```

*** Network Interface Configurations ***

Currently configured IP interfaces on the machine : 10.8.1.205

1 ) 10.1.1.111
2 ) 123.1.2.33

p ) Go back to previous menu
n ) Proceed to next configuration
a ) Abort Installation

Enter your choice : n

```



**Important:** Note that the script has sensed both the number of interfaces and their IP addresses.

**Step 7** Enter a line number to change a parameter value, if needed. This can be repeated as often as necessary to change parameter values. Then enter **n** to save changes (if made) or defaults.

Pressing **n** completes the menu-driven portion of the installation process and displays the configuration that you have created.

```

=====
Standalone Mode Installation
=====

GSS installation path : /<install_dir>/gss

*** PostgreSQL Configurations ***

PostgreSQL port : 5432

```

```

PostgreSQL login : postgres
PostgreSQL passwd : postgres
**** GSS Configurations ****
File Format for data files : starent
Hard Limit Interval for File Generation : 0
GTPP Dictionary : custom1
Support for LRSN rewrite : n []
Encoding of IP Address in binary format : n
GSN Location : GSN
Enable redundant data file support : n
*** Network Host Configurations ***
IP Address of the machine to be used : 10.1.1.111
=====

```

You are given the opportunity to modify the configuration that you have created. (The values displayed above were entered for illustration and not as recommendations.)

```
Do you want to Modify Configuration [n] ? n
```

**Step 8** Press **y** to return to the menus and change the configuration or press **n** to continue the installation process.

```

Installing GSS..... Please wait.....

Extracting perl tar... Done.

Add following entry to crontab (if not already present) to remove
processed data files in /<install_dir>/gss/data after storage period of 7
day(s)

0 * * * * /<install_dir>/gss/bin/cleanup.sh >>
/<install_dir>/gss/log/cleanup.log 2>&1

To start Process Monitor Tool along with GSS and Filegen (if not started
already) : execute "/<install_dir>/gss/serv start"

To get help on "GSS": execute "/<install_dir>/gss/serv help"

For additional info and performance tuning please read README & GSS User
Guide in doc directory

Do You Want To Start GSS : [y] ? y

```

This is the last action that you must take to complete the installation process.

**Step 9** Enter **n** to complete the installation without starting the GSS. Press return to accept the **yes** default to complete the installation process and start the GSS.

```
This will start Process Monitor Tool along with GSS and Filegen using
params listed in /<install_dir>/gss/etc/gss.cfg
```

```
Please see log/psmon.log file for log messages
```

```
Starting Process Monitor Tool...
```

```
Done.
```

```
Capturing status, please wait for a while...
```

```
=====
```

```
0 1118 12:04:16 TS 59 0:00 <install_dir>/gss/bin/gssfilegen 1
```

```
0 1015 12:04:10 TS 59 0:00 /usr/bin/bash <install_dir>/gss/serv start
29661
```

```
0 1102 12:04:15 TS 59 0:00 <install_dir>/gss/lib/perl5.8.5/bin/perl
<install_dir>/gss/psmon --daemon --cro 1
```

```
0 1113 12:04:16 TS 59 0:00 <install_dir>/gss/bin/gss 1
```

```
=====
```

```
GTPP Storage Server Version 8.0.xx installation done.
```

The status display indicates that GSS, FileGen, PSMON, and PostgreSQL have all been started. If nothing displays, turn to the *Troubleshooting the GSS* section in the *GTPP Storage Server Administration* chapter. In most cases, if the other components are started, then the PostgreSQL has also been started.

**Step 10** Optional: Confirm the location of the installed GSS components.

```
# cd /<install_dir>
```

```
# ls -al
```

```
total 16
```

```
drwxr-xr-x 4 root other 512 Dec 9 17:29 .
```

```
drwxr-xr-x 11 root root 512 Dec 9 17:28 ..
```

```
-rwxrwxrwx 1 root other 3782 Dec 9 17:28 StandaloneGSSUninstall.sh
```

```
drwxrwxr-x 12 root root 512 Dec 9 17:29 gss
```

```
drwxr-xr-x 6 postgres other 512 Dec 9 17:28 postgres
```

**Step 11** Optional: Check the location of the GSS application files, tools and engineering documentation.

```
# cd gss
```

```
# ls -al
total 232
drwxrwxr-x 12 root root 512 Dec 21 22:43 .
drwxr-xr-x 4 root other 512 Dec 21 22:43 ..
-rw-r--r-- 1 postgres other 533 Aug 17 10:58 .configfile
-r--r--r-- 1 postgres root 1168 Aug 17 10:58 .gss.env
-rw----- 1 root other 5 Dec 21 22:43 .gss.pid
-rw----- 1 root other 5 Dec 21 22:43 .gssfilegen.pid
-rw----- 1 root other 7 Dec 21 22:43 .gssfilegen.seq
-rw-r--r-- 1 root other 0 Dec 21 22:42 .inst_serv.err
-rwxr-xr-x 1 root root 4057 Mar 3 2006 README
drwxrwxr-x 2 root root 512 Dec 21 22:42 bin
drwxrwxr-x 2 root root 512 Sep 8 2004 data
drwxrwxr-x 2 root root 512 May 31 2005 doc
drwxrwxr-x 2 root root 512 Dec 21 22:42 etc
-rwxr-xr-x 1 root other 22480 Dec 21 22:42 gss_ctl
drwxrwxr-x 3 root root 512 Dec 21 22:42 lib
drwxrwxr-x 3 root root 512 Dec 21 22:43 log
-rwxr-xr-x 1 root other 54445 Dec 21 22:42 psmon
-rw-r--r-- 1 root other 4 Dec 21 22:43 psmon.pid
-rwxr-xr-x 1 root other 19850 Dec 21 22:42 serv
drwxrwxr-x 2 root root 512 Oct 10 05:43 sql
drwxrwxr-x 2 root root 512 Oct 10 05:43 template
drwxrwxr-x 2 root root 512 Sep 8 2004 tmp
drwxrwxr-x 3 root root 512 Dec 21 22:42 tools
```

**Step 12** Optional: Locate the GSS configuration file (gss.cfg).

```
# cd etc
# ls -al
total 42
```

```
drwxrwxr-x 2 root root 512 Dec 21 22:42 .
drwxrwxr-x 12 root root 512 Dec 21 22:43 ..
-rw-r--r-- 1 root other 10921 Dec 21 22:42 gss.cfg
-rw-r--r-- 1 root other 2459 Dec 21 22:42 gsslogger.xml
-rw-r--r-- 1 root other 3690 Dec 21 22:42 psmon.cfg
-rw-r--r-- 1 root other 261 Dec 21 22:42 uninstall_config_file
```

Before working with the GSS, it is recommended to create a write-protected copy of the *gss.cfg* file and store it in a separate directory. To ensure you remember the configuration for your software version, we suggest that you store the file in the */<packages>/<gss\_version>* directory.

## Installing the Complete GSS - Node 1 in Cluster

This section describes the process for installing the GSS server application, and all of the associated GSS components, on the primary GSS node of the cluster.

Prior to installing the GSS application, ensure that the cluster is installed and configured as needed. For information on installing and configuring the Sun cluster, refer to the *Sun documentation*.

### Using the Installation Script


Installation is accomplished using the `inst_serv` script. It provides a menu-driven interface with question prompts. Most prompts display default values or information derived from the server's current setup - such as IP addresses for configured interfaces.

The following information will help you use the installation script most effectively:

- **Ctrl-C** will abort the installation process at any time during the procedure.
- The script senses whether the server is a stand-alone node or a cluster node.
- To accept defaults and continue the process, enter **n** as your choice for any prompt.
- To change a displayed value, enter the menu item number to display an entry field or a list of possible values.

The information from the prompts is used to generate the GSS configuration file (`gss.cfg`). This file can be changed at anytime after the installation.

---

 **Important:** It is recommended that you fill in path prompts only after you have created the directories to be used.

---

### Installation Procedure - Node 1


The following procedure assumes that you are logged in to the GSS server with root privileges and that you are starting from the root directory.

**Step 1** Change to the `</packages>/gss_<version>` directory where you stored the GSS application software.

**Step 2** Locate the installation script file `inst_serv` and execute the following command:

```
./inst_serv
```

---

 **Important:** This script will check the version of operating system and cluster software installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Cluster Deployment* section, the script will abort the GSS installation.

---

The following appears, with pauses for validation, after entering the `inst_serv` command.

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
required changes in '/etc/system' file, restart the machine to get these
changes reflected and then start installation again.
```

```
set msgsys:msginfo_msgmnb=65536
```

```
set msgsys:msginfo_msgtql=1024
```

```
set shmsys:shminfo_shmmax=33554432
```

```
set shmsys:shminfo_shmmin=1
```

```
set shmsys:shminfo_shmmni=256
```

```
set shmsys:shminfo_shmseg=256
```

```
set semsys:seminfo_semmap=256
```

```
set semsys:seminfo_semmni=512
```


```
set semsys:seminfo_semmns=512
```

```
set semsys:seminfo_semmsl=32
```

```
Using cluster for Installation
```

```
Cluster Mode Installation (To be used in cluster environment) [n] ? y
```

---

 **Important:** Note that the script senses whether the server is a Stand-alone node or a Cluster node. If you made changes to the system file as in the [Installation First Steps](#) section, then you can continue. If not, abort the installation using CTRL+C, make the changes to the system file, and then reboot. After rebooting, begin the installation procedure again.

---

**Step 3** Enter **y(yes)** to continue the installation.

```
GTPP Storage Server installation directory


Specify the common path for GSS data and logs on all cluster nodes
[/sharedgss/gss] ?
```

**Step 4** Press **Enter** to accept the default directory `/sharedgss/gss` or enter the name of another directory. Next you are prompted for the location to install the GSS.

```
Path where Gss will be installed :

GSS Installation dir ? /TEST_GSS/cvserver
```

---

 **Important:** In case of cluster mode, it is highly recommended that you do not install the GSS application in the `/opt`, `/opt/gss/`, or `/opt/postgres` directory.


---

**Step 5** Press **Enter** to accept the default or enter the name of the directory where the GSS active components are to be installed. It is recommended that you put this directory at the root level. The installation script creates the directory if needed.

Shortly after responding to the prompt for the installation directory, the following appears:

```
Do you want Backup installation for current cluster mode installation
[y/n]: [n] ?
```

---

 **Important:** This enables backup mode for the GSS node in a cluster deployment.

---

If you do not want the backup for cluster mode installation, proceed to *step 7*. Otherwise, continue with next step.

**Step 6** Enter **y(yes)** to enable node switchover. The installation continues with a menu to configure the PostgreSQL parameters for backup.

```
*** PostgreSQL configuration for backup Installation ***

1) PostgreSQL port for backup installation :

2) PostgreSQL login for backup installation :

3) PostgreSQL data directory for backup installation :

n) Proceed to next configuration

a) Abort Installation

Enter Your Choice : [n] ?
```

**Step 7** Enter the number or letter of your choice and enter the postgres details as needed. The installation continues with a menu to configure the PostgreSQL parameters.

Entering n will save configuration values and take you to next configuration. To change the default values, enter option number

\*\*\* PostgreSQL installation configuration \*\*\*

- 1) PostgreSQL port : 5432
- 2) PostgreSQL login : postgres
- 3) PostgreSQL passwd : postgres
- 4) Shared PostgreSQL dir : /sharedpostgres
- n) Proceed to next configuration
- a) Abort Installation

Enter Your Choice : [n] ?

**Step 8** Enter the number or letter of your choice. Make changes as needed.

Entering n will save configuration values and take you to next configuration. To change the default values, enter option number

**Step 9** Enter n to save changes or defaults and move to the next menu.



**Important:** The GSS and FileGen are set to run in archive mode by default. The archive mode, used in deployments that do not include CGFs, instructs the server to save records to a file.

\*\*\* GSS Configuration Parameters \*\*\*

- 1) File Format for data files : starent
- 2) Hard Limit Interval for File Generation (mins) : 0
- 3) Support for LRSN rewrite : n
- 4) Encoding of IP Address in binary format : n
- 5) Enable redundant data file support : n
- 6) GSN Location : GSN
- 7) Specify the GTPP Dictionary : custom6
- p) Go back to previous menu
- n) Proceed to next configuration
- a) Abort Installation



**Important:** The **Specify the GTPP Dictionary** option appears only if the **Support for LRSN rewrite** or **Encoding of IP Address in binary format** parameter is enabled. Otherwise, the GTPP dictionary is set to default.

**Step 10** Enter the number or letter of your choice. Make changes as needed. Then enter **n** to save changes or defaults and move to the next menu.

```
*** Network Interface Configuration ***

1) Logical Host IP Addresss :
2) Logical Host Name :
3) Additional Logical Host Name[eg.For Mediation Server] : n
p) Go back to previous menu
n) Proceed to next configuration
a) Abort Installation

Enter your choice : n
```

**Step 11** Enter option 1.

```
Enter Your Choice :
1

Please specify already available logical host Address for GSS cluster : ?
```

**Step 12** Enter the IP address of the logical host, press **Enter**, and move to the next prompt.

```
Enter Your Choice :
```

**Step 13** Enter option 2.

```
Please specify Logical hostname for above logical host address : ?
```

**Step 14** Enter a line number to change any parameter value, if needed. This can be repeated as often as necessary to change parameter values. Then enter **n** to save changes (if made) or defaults.

Pressing **n** completes the menu-driven portion of the installation process and displays the configuration that you have created.

```
=====
Cluster Mode Installation
=====

GSS installation path : /TEST_GSS/cvserver/gss
Common path for GSS data and logs : /sharedgss/gss

*** Backup PostgreSQL Configurations ***
```

```

PostgreSQL port for backup installation : 5477
PostgreSQL login for backup installation : backpost
PostgreSQL data directory for backup installation : /backpost
*** PostgreSQL Configurations ***
PostgreSQL port : 5432
PostgreSQL login : gsspg
PostgreSQL passwd : gsspg
Shared PostgreSQL dir : /sharedpostgres
**** GSS Configurations ****
File Format for data files : custom7
Hard Limit Interval for File Generation : 2
GTPP Dictionary : custom6
Support for LRSN rewrite : n []
Encoding of IP Address in binary format : n
GSN Location : GSN
Enable redundant data file support : n
*** Network Host Configurations ***
Logical Host IP Addresss : 10.1.1.1
Logical Host Name : gssserv
Additional Logical Host Name : n
=====

```

You are given the opportunity to modify the configuration that you have created.

```
Do you want to Modify Configuration [y/n] : [n] ?
```

**Step 15** Press **y** to return to the menus and change the configuration or press **n** or **Enter** to continue the installation process.

```
Installing GSS..... Please wait.....
```

```
Extracting perl tar...
```

```
Done.
```

```
Add following entry to crontab (if not already present) to remove
processed data files in /sharedgss/gss/data after storage period of 7
day(s)
```

```

0 * * * * /TEST_GSS/cvserver/gss/bin/cleanup.sh >>
/sharedgss/gss/clustems1_log//cleanup.log 2>&1

For additional info and performance tuning please read README & GSS User
Guide in doc directory

GSS Cluster Agent Installation configuration

Proceeding with GSS Cluster Agent Installation

Extracting StarentGss.tar...

Processing package instance <StarentGss> from </TEST_GSS/gss_x_x_xx>

Sun Cluster resource type for Gss server(sparc) 3.0.0,REV=
Sun Microsystems, Inc.

Using </opt> as the package base directory.

## Processing package information.
## Processing system information.
## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <StarentGss> [y,n,?] y

```

**Step 16** Enter **y(yes)** to continue the installation.

```

Installing Sun Cluster resource type for Gss server as <StarentGss>

## Installing part 1 of 1.

/opt/StarentGss/README.Gss

/opt/StarentGss/bin/Gss_mon_check.ksh

/opt/StarentGss/bin/Gss_mon_start.ksh

/opt/StarentGss/bin/Gss_mon_stop.ksh

/opt/StarentGss/bin/Gss_probe.ksh

/opt/StarentGss/bin/Gss_svc_start.ksh

/opt/StarentGss/bin/Gss_svc_stop.ksh

```

```

/opt/StarentGss/bin/Gss_update.ksh
/opt/StarentGss/bin/Gss_validate.ksh
/opt/StarentGss/bin/gethostnames
/opt/StarentGss/bin/gettime
/opt/StarentGss/bin/hasp_check
/opt/StarentGss/bin/simple_probe
/opt/StarentGss/etc/Starent.Gss
/opt/StarentGss/man/man1m/Gss_config.1m
/opt/StarentGss/man/man1m/removeGss.1m
/opt/StarentGss/man/man1m/startGss.1m
/opt/StarentGss/man/man1m/stopGss.1m
/opt/StarentGss/util/Gss_config
/opt/StarentGss/util/removeGss
/opt/StarentGss/util/startGss
/opt/StarentGss/util/stopGss
[ verifying class <none> ]
## Executing postinstall script.
Installation of <StarentGss> was successful.
GTPP Storage Server Version x.x.xx installation done.
GSS installation on node1 is complete. please proceed with installation
on node2....

```

## Installing the Complete GSS - Node 2 in Cluster

This section describes the process for installing the GSS server application, with all of the GSS components, on to a second GSS node of a cluster.

The following procedure assumes that you are logged in to the node 2 (standby) GSS server with root privileges and that you are starting from the root directory

- Step 1** Change to the `<packages>/gss_<version>` directory where you stored the GSS application software.
- Step 2** Locate the installation script file `inst_serv` and execute the following command:

`./inst_serv`



**Important:** This script will check the version of operating system and cluster software installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Cluster Deployment* section, the script will abort the GSS installation.

The following appears, with pauses for validation, after entering the `inst_serv` command.

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
required changes in '/etc/system' file, restart the machine to get these
changes reflected and then start installation again.
```

```
set msgsys:msginfo_msgmnb=65536
```

```
set msgsys:msginfo_msgtql=1024
```

```
set shmsys:shminfo_shmmax=33554432
```

```
set shmsys:shminfo_shmmin=1
```

```
set shmsys:shminfo_shmmni=256
```

```
set shmsys:shminfo_shmseg=256
```

```
set semsys:seminfo_semmap=256
```

```
set semsys:seminfo_semmni=512
```

```
set semsys:seminfo_semmns=512
```

```
set semsys:seminfo_semmsl=32
```

```
Using cluster for Installation
```

```
Cluster Mode Installation (To be used in cluster environment)[y/n]: [n] ?
```

**Step 3** Enter **y(yes)** to perform GSS installation in cluster mode and continue the installation.

```
GTPP Storage Server installation directory
```

Specify the common path for GSS data and logs on all cluster nodes  
 [/sharedgss/gss] ?

**Step 4** Enter the name of the directory where files that will be common to all nodes in the cluster are to be stored, for example /<shared\_gss\_dir>. You must enter the same path that you entered when you installed the GSS on node 1.

Shortly after typing /<shared\_gss\_dir> and pressing **Enter**, the configuration created for cluster node 1 appears:

```

=====
Cluster Mode Installation
=====
GSS installation path : /TEST_GSS/cvserver/gss
Common path for GSS data and logs : /sharedgss/gss
*** Backup PostgreSQL Configurations ***
PostgreSQL port for backup installation : 5477
PostgreSQL login for backup installation : backpost
PostgreSQL data directory for backup installation : /backpost
*** PostgreSQL Configurations ***
PostgreSQL port : 5499
PostgreSQL login : gsspg
PostgreSQL passwd : gsspg
Shared PostgreSQL dir : /sharedpostgres
**** GSS Configurations ****
File Format for data files : custom7
Hard Limit Interval for File Generation : 2
GTPP Dictionary : custom6
Support for LRSN rewrite : n []
Encoding of IP Address in binary format : n
GSN Location : GSN
Enable redundant data file support : n
*** Network Host Configurations ***
Logical Host IP Addresss : 10.4.72.110
Logical Host Name : gssserv

```

Additional Logical Host Name : n

=====

If you want to change configuration values, do not start GSS after installation. Change required configuration parameters from /TEST\_GSS/cvserver/gss/etc/gss.cfg and then start GSS..

Proceed With Installation [y/n] : [y] ?

**Step 5** Press **Enter** to continue the installation.

Installing GSS..... Please wait.....

Extracting perl tar... Done.

Add following entry to crontab (if not already present) to remove processed data files in /sharedgss/gss/data after storage period of 7 day(s)

```
0 * * * * /TEST_GSS/cvserver/gss/bin/cleanup.sh >>
/sharedgss/gss/clustems2_log//cleanup.log 2>&1
```

For additional info and performance tuning please read README & GSS User Guide in doc directory

GSS Cluster Agent Installation configuration

Proceeding with GSS Cluster Agent Installation

Extracting StarentGss.tar...

Processing package instance <StarentGss> from </TEST\_GSS/gss\_8\_1\_dict21>

Sun Cluster resource type for Gss server(sparc) 3.0.0,REV=

Sun Microsystems, Inc.

Using </opt> as the package base directory.

## Processing package information.

## Processing system information.

## Verifying package dependencies.

## Verifying disk space requirements.

## Checking for conflicts with packages already installed.

## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user permission during the process of installing this package.

Do you want to continue with the installation of <StarentGss> [y,n,?]

**Step 6** Enter **y (yes)** to continue the installation.

```
Installing Sun Cluster resource type for Gss server as <StarentGss>
## Installing part 1 of 1.
/opt/StarentGss/README.Gss
/opt/StarentGss/bin/Gss_mon_check.ksh
/opt/StarentGss/bin/Gss_mon_start.ksh
/opt/StarentGss/bin/Gss_mon_stop.ksh
/opt/StarentGss/bin/Gss_probe.ksh
/opt/StarentGss/bin/Gss_svc_start.ksh
/opt/StarentGss/bin/Gss_svc_stop.ksh
/opt/StarentGss/bin/Gss_update.ksh
/opt/StarentGss/bin/Gss_validate.ksh
/opt/StarentGss/bin/gethostnames
/opt/StarentGss/bin/gettime
/opt/StarentGss/bin/hasp_check
/opt/StarentGss/bin/simple_probe
/opt/StarentGss/etc/Starent.Gss
/opt/StarentGss/man/man1m/Gss_config.1m
/opt/StarentGss/man/man1m/removeGss.1m
/opt/StarentGss/man/man1m/startGss.1m
/opt/StarentGss/man/man1m/stopGss.1m
/opt/StarentGss/util/Gss_config
/opt/StarentGss/util/removeGss
/opt/StarentGss/util/startGss
/opt/StarentGss/util/stopGss
[ verifying class <none> ]
## Executing postinstall script.
Installation of <StarentGss> was successful.
Do you want to start GSS on cluster nodes [y/n] : [y] ?
```

**Step 7** Enter **y (yes)** to continue the installation.

```
Starting GSS on cluster node....
```

```
Capturing status, please wait for a while...
```

```
=====
```

```
0 6523 00:17:37 TS 59 00:01
/TEST_GSS/cvserver/gss/lib/perl5.8.5/bin/perl
/TEST_GSS/cvserver/gss/psmon --da 1

0 6539 00:17:38 TS 59 00:00 /TEST_GSS/cvserver/gss/bin/gssfilegen 1

0 6534 00:17:38 TS 59 00:00 /TEST_GSS/cvserver/gss/bin/ gss 1

gsspg 6549 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6486 1 0 00:17:32 ? 0:00 /TEST_GSS/cvserver/postgres/bin/postgres
-D /TEST_GSS/cvserver/postgres/data

gsspg 6552 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6548 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6545 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6560 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6550 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6562 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6500 6486 0 00:17:32 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6498 6486 0 00:17:32 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data
```

```
gsspg 6558 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6543 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6554 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6559 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6556 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6551 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6561 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6565 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6501 6486 0 00:17:32 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6540 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6566 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6553 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data

gsspg 6564 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data
```

```
gsspg 6555 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data
```

```
gsspg 6546 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data
```

```
gsspg 6563 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data
```

```
gsspg 6557 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data
```

```
gsspg 6544 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data
```

```
gsspg 6547 6486 0 00:17:38 ? 0:00
/TEST_GSS/cvserver/postgres/bin/postgres -D
/TEST_GSS/cvserver/postgres/data
```

```
=====
```

If there is any problem with cluster mode GSS, you can start backup mode GSS using `"/TEST_GSS/cvserver/gss/serv start_backup"`.

GTPP Storage Server Version x.x.xx installation done.

The status display indicates that GSS, FileGen, PSMON, and Postgres have all been started. If nothing displays, turn to the *Troubleshooting the GSS* section in the *GTPP Storage Server Administration* chapter. In most cases, if the other components started, then the PostgreSQL has also been started.

**Step 8** Optional. Return to the command line of the primary node. Confirm that the PostgreSQL engine has started by entering the following command:

```
ps -ef
```

The resulting display will display all the processes running and there will most likely be multiple lines that confirm the PostgreSQL is running as they look similar to the following:

```
postgres 15080 14972 0 07:23:25 ? 0:00
/<clus_install_dir>/postgres/bin/postgres -D
```

**Step 9** Optional. Locate the GSS configuration file (`gss.cfg`).

```
# cd etc
# ls -al
total 38
drwxr-xr-x 2 1460 100 512 Apr 20 07:22 .
```

```

drwxr-xr-x 12 1460 100 512 Apr 20 07:23 ..
-rw-r--r-- 1 root other 12561 Apr 20 07:22 gss.cfg
-rw-r--r-- 1 root other 3521 Apr 20 07:22 psmon.cfg

```

Before doing anything with the GSS, it is recommended to create a write-protected copy of the *gss.cfg* file and store it in a separate directory. To ensure you remember the configuration for your software version, we suggest that you store it in the */<packages>/<gss\_version>* directory.

## Uninstalling the Complete GSS - Stand-alone Node

Uninstalling the GSS accomplishes the following:

- Deletes gss directory from GSS installation path.
- Deletes postgres directory from postgres installation path
- Deletes gss, postgres and bin directories from GSS installation path.
- Deletes the postgres user created.
- Cleans shared memory and semaphores for the given postgres user.
- Removes postgres lock files from /tmp for given postgres port.
- Removes GSS startup scripts from /etc directory (rc\*.d and init.d)
- Removes Postgres startup scripts from /etc directory (rc\*.d and init.d)

Uninstalling is easily performed using the node-specific script and configuration file that are generated during installation. During GSS installation, these files are inserted into the generated directory where the GSS application files are stored.

**Step 1** Change to the directory where the GSS application files were installed.

```
# cd /<install_dir>
```

**Step 2** List the files in the directory. Find the script (**extension .sh**).

```

# cd install_8071

# ls -al

total 16

drwxr-xr-x 4 root other 512 Dec 21 22:43 .
drwxr-xr-x 11 root root 512 Dec 21 22:42 ..
-rwxrwxrwx 1 root other 3782 Dec 21 22:42 StandaloneGSSUninstall.sh
drwxrwxr-x 12 root root 512 Dec 21 22:43 gss

drwxr-xr-x 6 postgres other 512 Dec 21 22:42 postgres

```

**Step 3** Enter the name of the uninstall script to begin the uninstall process.

```
# ./StandaloneGSSUninstall.sh
```

```
Uninstallation Process will Remove Installation directories
```

```
Do You Want To Proceed With Uninstallation [y/n] : [n] ? y
```

**Step 4** Enter **y(yes)** to continue the uninstall.

The system goes to the installation directory where the system-generated `uninstall_config_file` resides and uses the information in that file to complete the install process.

```
Uninstalling </<install-dir>/gss/etc/uninstall_config_file...
This will stop Process Monitor Tool along with GSS and Filegen
Please see log/psmon.log file for log messages
Stopping Process Monitor Tool...
Done.
Stopping GSS...
Done.
Stopping GSS FileGen...
Done.
waiting for server to shut down.... done
server stopped
starting cleanup process
Cleaning the gss installation paths ...
Cleaning the postgres installation paths ...
Cleaning the shared memory and semaphores ...
Deleting the postgres user created during installation ...
Cleaning the postgres lock from /tmp directory ...
Uninstallation process completed successfully.....#
```

## Uninstalling the Complete GSS - Cluster Nodes

Uninstalling the GSS from a cluster node accomplishes the following:

- Deletes the `gss` directory from GSS installation path.
- Deletes the `postgres` directory from the Postgres installation path.

- Deletes gss, postgres and bin directories from GSS installation path.
- Deletes the Postgres user created during installation.
- Cleans the shared memory and semaphores for the given \_Postgres user.
- Removes Postgres lock files from /tmp for given postgres port.
- Removes GSS startup scripts from /etc directory (rc\*.d and init.d)
- Removes Postgres startup scripts from /etc directory (rc\*.d and init.d)
- Stops the GSS and Postgres processes for the node.
- Brings cluster offline from both the nodes.
- Disables CRNP, gss resource, network resource and addition resource (if defined).
- Unmanages the Gss-harg resource group.
- Deletes CRNP, GSS, network resource and additional resource.
- Un-registers CRNP resource type, GSS resource type and finally removes the Gss-harg resource group.
- Deletes the gss directory from gss shared data path and postgres directory from postgres shared data path.
- Removes the StarentGss package from the server.

During both the installation and the upgrade processes, the system generates system-specific uninstall files: an uninstall script and an uninstall configuration file.

Begin with the primary node 1.

**Step 1** Change to the directory where the GSS application files were installed.

```
# cd /<clus_install_dir>
```

**Step 2** List the files in the directory. Find the script (extension .sh).

```
# cd install_8071
# ls -al

total 16

drwxr-xr-x 4 root other 512 Dec 21 22:43 .
drwxr-xr-x 11 root root 512 Dec 21 22:42 ..
-rwxrwxrwx 1 root other 3782 Dec 21 22:42 ClusterGSSUninstall.sh
drwxrwxr-x 12 root root 512 Dec 21 22:43 gss
drwxr-xr-x 6 postgres other 512 Dec 21 22:42 postgres
```

**Step 3** Enter the name of the uninstall script to begin the uninstall process.

```
bash-2.05# ./ClusterGSSUninstall.sh

Uninstallation Process will remove shared directories, installation
directories....

Do You Want To Proceed With Uninstallation [y/n] : [n] ?
```

**Step 4** Enter **y (yes)** to continue the uninstall and start the cleanup operation.

```
Gss-harg: invalid resource group
starting cleanup process
checking if resource group exists..
Cleaning the gss and postgres common data paths ...
Cleaning the starentGss package ...
The following package is currently installed:
  StarentGss Sun Cluster resource type for Gss server
  (sparc) 3.0.0,REV=
Do you want to remove this package? [y,n,?,q]
```

**Step 5** Enter **y (yes)** to continue.

```
## Removing installed package instance <StarentGss>
This package contains scripts which will be executed with super-user
permission during the process of removing this package.
Do you want to continue with the removal of this package [y,n,?,q]
```

**Step 6** Enter **y (yes)** to continue.

```
## Verifying package dependencies.
## Processing package information.
## Executing preremove script.
Resource <Gss-hars> has been removed already
Resource type <Starent.Gss> has been un-registered already
Network Resource not removed...
You may run removeGss again with the -h option to remove network
resource.
## Removing pathnames in class <none>
/opt/StarentGss/util/stopGss
/opt/StarentGss/util/startGss
/opt/StarentGss/util/removeGss
/opt/StarentGss/util/Gss_config
/opt/StarentGss/util
```

```
/opt/StarentGss/man/man1m/stopGss.1m
/opt/StarentGss/man/man1m/startGss.1m
/opt/StarentGss/man/man1m/removeGss.1m
/opt/StarentGss/man/man1m/Gss_config.1m
/opt/StarentGss/man/man1m
/opt/StarentGss/man
/opt/StarentGss/etc/Starent.Gss
/opt/StarentGss/etc
/opt/StarentGss/bin/simple_probe
/opt/StarentGss/bin/hasp_check
/opt/StarentGss/bin/gettime
/opt/StarentGss/bin/gethostnames
/opt/StarentGss/bin/Gss_validate.ksh
/opt/StarentGss/bin/Gss_update.ksh
/opt/StarentGss/bin/Gss_svc_stop.ksh
/opt/StarentGss/bin/Gss_svc_start.ksh
/opt/StarentGss/bin/Gss_probe.ksh
/opt/StarentGss/bin/Gss_mon_stop.ksh
/opt/StarentGss/bin/Gss_mon_start.ksh
/opt/StarentGss/bin/Gss_mon_check.ksh
/opt/StarentGss/bin
/opt/StarentGss/README.Gss
/opt/StarentGss
## Executing postremove script.
## Updating system information.
Removal of <StarentGss> was successful.
Cleaning the gss installation paths
Cleaning the postgres installation paths ...
Cleaning the /opt directories paths ...
```

```

Cleaning the shared memory and semaphores

Deleting the postgres user created during installation ...

Cleaning the postgres lock from /tmp directory

Uninstallation process completed successfully.....

root@clustgss2#

```

**Step 7** Login to node 2 and repeat *step 1* through *step 6* to uninstall GSS from the standby node.

## Upgrading the GSS Stand-alone Node

This process upgrades both the GSS and Postgres database application files on a stand-alone node. The script begins by stopping all active processes and then restarts them after the upgrade is completed.

This upgrade process is valid for upgrading:

- GSS servers loaded with 7.1.x to GSS 8.0.x, or
- GSS servers loaded with 7.1.x to GSS 8.1.x.

## Preparing to Upgrade

**Step 1** Login to the GSS server.

**Step 2** Copy the current *gss.cfg*.

- Change to `/<clus_install_dir>/gss/etc` and make a copy of the current *gss.cfg*.
- Rename the file and store it in a separate directory. It is recommended to store the copy in the `/<packages>/<gss_version>` directory holding the current GSS version.

**Step 3** Move a copy of the new GSS version to the directory where you have stored other folders containing earlier version. Unzip the compressed file for the new GSS version by entering the following command:

```
gunzip gss_<version>.tar.gz
```

`<version>` is the version number of the GSS software distributed in the zip file. For example, *gss\_8.0.xx.tar.gz*

**Step 4** Locate the tar file *GSS\_<version>.tar* in the `/<packages>` directory and untar the file by entering the following command:

```
tar -xvf gss_<version>.tar
```

During the untar process, a `/gss_<version>` directory (for example: `/gss_8.0.xx`) is created in the `/<packages>` directory.

## Using the Installation Script


Installation is accomplished using the **inst\_serv** script. It provides a menu-driven interface with question prompts. Most prompts display default values or information derived from the server's current setup - such as IP addresses for configured interfaces.

The following information will help you use the installation script most effectively:

- **Ctrl-C** will abort the installation process at any time during the procedure.
- The script senses whether the server is a stand-alone node or a cluster node.
- To accept defaults and continue the process, enter **n** as your choice for any prompt.
- To change a displayed value, enter the menu item number to display an entry field or a list of possible values.

The information from the prompts is used to generate the GSS configuration file (**gss.cfg**). This file can be changed at anytime after the installation.

---

 **Important:** It is recommended that you fill in path prompts only after you have created the directories to be used.

---

## Upgrading a GSS Stand-alone Node


The following procedure assumes that you are logged in to the GSS server with root privileges and that you are starting at the root directory level.

**Step 1** Change to the `/<packages>/gss_<version>` directory that was created when the files were uncompressed.

**Step 2** Locate the installation script file **inst\_serv** and execute the following command:

```
./inst_serv
```

---

 **Important:** This script will check the version of the operating system installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Stand-alone Deployment* section, the script will abort the GSS installation.

---

The following appears, with pauses for validation, after entering the **inst\_serv** command:

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
```

required changes in '/etc/system' file, restart the machine to get these changes reflected and then start installation again.

```
set msgsys:msginfo_msgmnb=65536
set msgsys:msginfo_msgtql=1024
set shmsys:shminfo_shmmax=33554432
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=256
set shmsys:shminfo_shmseg=256
set semsys:seminfo_semmmap=256
set semsys:seminfo_semmni=512
set semsys:seminfo_semmns=512
set semsys:seminfo_semmsl=32
```

Standalone Mode Installation [in standalone environment]...

GTPP Storage Server installation directory

Path where Gss will be installed :

GSS Installation dir ? /<install\_dir>



**Important:** If the upgrade is to a GSS 8.0.xx then /opt will appear as a default value for the **GSS Installation dir** parameter.

**Step 3** Enter the name of the directory where the new GSS is to be installed.

```
</install_dir>
```

**Step 4** After entering the name of the directory the script senses that GSS is already installed and prompts to upgrade:

```
Do you want to upgrade GSS [n] ? y
```

**Step 5** Enter **y (yes)** to continue the upgrade.

```
Starting Standalone upgrade
```

```
This will stop Process Monitor Tool along with GSS and Filegen
```

```
Please see log/psmon.log file for log messages
```

```
Stopping Process Monitor Tool...
```

```
Done.
```

```
Stopping GSS...
```

```

Done.

Stopping GSS FileGen...

Process GSS FileGen could not be stopped..

Killing it...

Done.

waiting for server to shut down.... done

server stopped

server starting

Starting database upgrade...

Database upgrade completed.....

waiting for server to shut down.... done

server stopped

Do You Want To Start GSS [y/n]: [y] ?

```

**Step 6** Enter **y** or press **Enter** to start the GSS and all related processes (e.g. the Process Monitor).

This will start Process Monitor Tool along with GSS and Filegen using params listed in `<install_dir>/gss/etc/gss.cfg`

Please see `log/psmon.log` file for log messages

Starting Process Monitor Tool...

Done.

Capturing status, please wait for a while...

```

=====
0 17453 11:35:15 TS 59 0:00 /<install_dir>/gss/lib/perl5.8.5
/bin/perl -w /<install_dir>/gss/psmon --daemon --cron 1
0 17460 11:35:16 TS 59 0:01 /<install_dir>/gss/bin/gss 1
0 17465 11:35:16 TS 59 0:01 /<install_dir>/gss/bin/gssfilegen 1
0 17108 11:33:08 TS 59 0:00 tee -a /<install_dir>/gss_7_1_67
/installation_log_<install_dir>_gss1 17067
0 2791 Oct_13 TS 59 1:01 /bin/bash /<install_dir>/gss/bin
/test_process.sh 1

```

```

0 17421 11:35:12 TS 59 0:00 tee -a /</install_dir>/gss_7_1_67/
installation_log_</install_dir>_solleims 17106

postgres 17479 17435 0 11:35:16 ? 0:00 /</install_dir>/postgres/
bin/postgres -D /</install_dir>/postgres/data -i

postgres 17485 17435 0 11:35:16 ? 0:00 /</install_dir>/postgres/
bin/postgres -D /</install_dir>/postgres/data -i

...

...

...

postgres 17491 17435 0 11:35:16 ? 0:00 /</install_dir>/postgres/
bin/postgres -D /</install_dir>/postgres/data -i

=====

GTPP Storage Server Version <gss_x_x_xx> installation done.

=====

```

The status display indicates that GSS, FileGen, PSMON, and PostgreSQL have all been started. If nothing displays, turn to the *Troubleshooting the GSS* section in the *GTPP Storage Server Administration* chapter.

## Upgrading the GSS - Cluster Nodes

This process upgrades both the GSS and Postgres database application files on the nodes in a cluster.

This upgrade process is valid for upgrading:

- GSS servers loaded with 7.1.x to GSS 8.0.x, or
- GSS servers loaded with 7.1.x to GSS 8.1.x

### Preparing to Upgrade

**Step 1** Login to the GSS server.

**Step 2** Copy the current *gss.cfg*.

- Change to /<clus\_install\_dir>/gss/etc directory and make a copy of the current *gss.cfg*.
- Rename the file and store it in a separate directory. It is recommended to store the copy in the /<packages>/<gss\_version> directory holding the current GSS version.

**Step 3** Move a copy of the new GSS version to the directory where you have stored other folders containing earlier version. Unzip the compressed file for the new GSS version by entering the following command:

```
gunzip gss_<version>.tar.gz
```

<version> is the version number of the GSS software distributed in the zip file. For example, *gss\_8.0.xx.tar.gz*

**Step 4** Locate the tar file *GSS\_<version>.tar* in the */<packages>* directory and untar the file by entering the following command:

```
tar -xvf gss_<version>.tar
```

During the untar process, a */gss\_<version>* directory (for example: */gss\_8.0.xx*) is created in the the */<packages>* directory.

## Upgrading Node1 - Primary Node

By using the upgrade procedure, you ensure that no CDRs are lost as there will always be one of the nodes in active/online mode.

The primary node in a cluster is typically referred to as node1. You should begin the process by logging in to the node1.

**Step 1** Change to the directory where the untarred files were stored and initiate the installation script.

```
./inst_serv
```



**Important:** This script will check the version of operating system and cluster software installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Cluster Deployment* section, the script will abort the GSS installation.

The following appears, with pauses for validation, after entering the **inst\_serv** command:

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
required changes in '/etc/system' file, restart the machine to get these
changes reflected and then start installation again.
```

```
set msgsys:msginfo_msgmnb=65536
```

```
set msgsys:msginfo_msgtql=1024
```

```

set shmsys:shminfo_shmmax=33554432
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=256
set shmsys:shminfo_shmseg=256
set semsys:seminfo_semmmap=256
set semsys:seminfo_semmni=512
set semsys:seminfo_semmns=512
set semsys:seminfo_semmsl=32

```

Cluster mode GSS installation exists /<server\_name>

**Step 2** The script registers that GSS is already installed and prompts you to upgrade the GSS installation. Enter **y** to upgrade the GSS application to a higher version.

```
Do you wish to upgrade this installation [y/n]: [n] ?
```

The following prompt appears only when the backup mode has not been configured during the fresh installation of GSS.

```
Do you want Backup installation for current cluster mode installation
[y/n]: [n] ?
```

```
*** PostgreSQL configuration for backup Installation ***
```

- 1) PostgreSQL port for backup installation :
- 2) PostgreSQL login for backup installation :
- 3) PostgreSQL data directory for backup installation :
- n) Proceed to next configuration
- a) Abort Installation

```
Enter Your Choice : [n] ?
```

**Step 3** After responding to continue the upgrade, the script displays upgrade requirements.

Please note that GSS upgrade in cluster mode needs following steps:-


1. Database upgrade on either of the nodes.
2. The node should be in standalone mode (after DB upgrade is done).

```
-----
```

This procedure will walk you through database upgrade first and then boot the node in standalone mode.

**Step 4** Indicate if you want to upgrade the database, press **Enter** or type **y**.

---

 **Important:** The database can only be upgraded on the node where the GSS is currently online (primary).

---

Do you want to upgrade database? (If you have already upgraded it on another node, you may answer no to this question and proceed to boot the node in standalone mode.) [y/n] : [y] ? y

The script proceeds to inform you of the next actions that are being taken.

Resource group "Gss-harg" is online on node "clustgss2".

For DB upgrade you need to start/switch resourcegroup on node clustgss1.

Do you want to start/switch resourcegroup "Gss-harg" on node clustgss1.  
[y/n] : [y] ?

Bringing resource group online on node clustgss1, please wait...

Using "cluster\_db\_upgrade" script from "<packages>/gss\_<version>" to upgrade database.

clustgss

Starting database upgrade...

Database upgrade completed successfully.....

You need to reboot the node in standalone mode to upgrade the GSS

After rebooting node in standalone mode, use "inst\_serv" script for GSS upgrade.

Do you want to reboot the node in standalone mode [y/n] : [y] ?

**Step 5** Press **Enter** or type y to reboot into stand-alone mode and continue the upgrade process.

Rebooting node .....

**Step 6** Login to node2 as root and confirm the switch of the primary node (node1) to standby and the standby (node2) to active mode.

login as: root

# clresourcegroup status

=== Cluster Resource Groups ===

Group Name	Node Name	Suspended	Status
-----	-----	-----	-----
Gss-harg	<name_node2>	No	Online
Gss-harg	<name_node1>	No	Offline

The display above indicates the upgrade script successfully performed the switchover.



**Important:** It will take a minute or two for the resource group to switch to node2 and for node1 to reboot in stand-alone mode.

**Step 7** Confirm node1 has rebooted in stand-alone mode by moving to node1 and entering the following command:

```
scstat
```

The system should indicate the node is not a cluster node:

```
scstate: not a cluster member.
```

**Step 8** Return to node1 and change to the directory where the GSS application files were stored, `/<packages>/gss_<version>` and initiate the installation script.

```
# ./inst_serv
```

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
required changes in '/etc/system' file, restart the machine to get these
changes reflected and then start installation again.
```

```
set msgsys:msginfo_msgmnb=65536
```

```
set msgsys:msginfo_msgtql=1024
```

```
set shmsys:shminfo_shmmax=33554432
```

```
set shmsys:shminfo_shmmin=1
```

```
set shmsys:shminfo_shmmni=256
```

```
set shmsys:shminfo_shmseg=256
```

```
set semsys:seminfo_semmap=256
```

```
set semsys:seminfo_semmni=512
```

```
set semsys:seminfo_semmns=512
```

```
set semsys:seminfo_semmsl=32
```

```
Cluster mode GSS installation exists /<clus_install_dir>
Do you wish to upgrade this installation [y/n]: [n] ?
```

**Step 9** Press **Enter** or type **y (yes)** to continue the upgrade.

```
Starting upgrade of cluster

Creating backup of previous installation files as
/<gss_install_dir>/gss_backup.tar

Backup done.

GSS Cluster Agent Installation configuration
Proceeding with GSS Cluster Agent Installation

Extracting StarentGss.tar...

Processing package instance <StarentGss> from </<package_dir>
/gss_<x_x_xx>

Sun Cluster resource type for Gss server (sparc) 3.0.0,REV=
Sun Microsystems, Inc.

This appears to be an attempt to install the same architecture and
version of a package which is already installed. This installation will
attempt to overwrite this package.

Using </opt> as the package base directory.

## Processing package information.
## Processing system information.

 28 package pathnames are already properly installed.

## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <StarentGss> [y,n,?] y
```

**Step 10** Press **Enter** or type **y** to confirm the upgrade of the cluster resources.

```
Installing Sun Cluster resource type for Gss server as <StarentGss>
## Installing part 1 of 1.
```

```

[ verifying class <none> ]
## Executing postinstall script.
Installation of <StarentGss> was successful.
Extracting perl tar... Done.
Starting Backup database upgrade...
Backup Database upgrade completed.....
waiting for server to shut down.... done
server stopped

You need to reboot the node in cluster mode for normal cluster mode
operation.

If database is not upgraded on another node, please run
"cluster_db_upgrade" script on another node for database upgrade.

Do you want to reboot the node in cluster mode [y/n] : [y] ?

```

- Step 11** Upgrade on node1 is complete so you need to confirm node1 reboot so that node will return to cluster mode. Press **Enter** or type **y (yes)** to reboot.

```
Rebooting node .....
```

- Step 12** Confirm node1 has returned to the cluster. Generate a status display with the following command:

```

cnode status

=== Cluster Nodes ===

--- Node Status ---

Node Name          Status
-----
<node_name1>      Online
<node_name2>      Online

```

The system recognizes node1 as the standby cluster node.

- Step 13** Confirm the version of GSS installed on node1 with the following command:


```
./GSS -version
```



**Important:** `./serv version` is used when upgrading to GSS versions higher than x.x.69.

- Step 14** Switch the GSS resource group from node2 back to node1 with the following commands found in the `/<install_dir>/gss` directory.


---

 **Important:** Reminder from previous step - `/opt` is a default installation directory for some versions of GSS.

---

```
cd /opt/gss
./GSS switch
```

---

 **Important:** `./serv switch` is used when upgrading to GSS versions higher than x.x.69.

---

After entering the switchover command, the system displays the following:

```
Resource group "Gss-harg" is online on node "<name_node2>".
Bringing resource group online on node "<name_node1>". Please wait ...
Done.
```

**Step 15** Confirm the switchover was successful, node1 is again the primary node, with the following command and display:

```
clresourcegroup status


=== Cluster Resource Groups ===

Group Name          Node Name          Suspended   Status
-----
Gss-harg            <name_node1>      No          Online
                   <name_node2>      No          Offline
```

Now GSS resource group is active with the new release and node2 is in standby mode and free to be upgraded.

**Step 16** Perform *step 1* to *step 11* on node2 to upgrade it to the newer version of GSS.

---


 **Important:** While executing *step 3* on the second node, do not choose to upgrade database, as it has already upgraded for first node. After completing the steps on node2, GSS upgrade on cluster setup will be completed.

---

## Multiple Instances of GSS

This section includes procedures for installation, uninstallation and upgrade to support multiple instances of GSS on a same cluster setup and standalone mode.

---

 **Important:** If you are deploying multiple instances of GSS, please refer to the procedures given below. From release 9.0 onwards, deployments of multiple instances of GSS must be used only on Solaris 10.

---

- [Installing multiple GSS Instances - Stand-alone Node](#)
- [Installing multiple GSS Instances - Node 1 in Cluster](#)
- [Installing multiple GSS Instances - Node 2 in Cluster](#)
- [Uninstalling Multiple GSS Instances - Stand-alone Node](#)
- [Uninstalling multiple GSS Instances - Cluster Nodes](#)
- [Upgrading multiple GSS Instances - Stand-alone Node](#)
- [Upgrading multiple GSS instances - Cluster Nodes](#)

The GSS installer installs all the binaries for GSS, PostgreSQL and StarentGSS (cluster resource type) package in */opt/gss\_global* directory. The installer has to be executed on both the nodes of the cluster. The */opt/gss\_global/make\_gss\_instance* script utility is used to create instances after initial installation. Each instance will have a separate configuration file, log directory, and tools directory. During uninstallation, each instance can be uninstalled separately and will not have any impact on the other instances. Global installation can be only uninstalled if there are no instances configured or running on the system.

## Installing multiple GSS Instances - Stand-alone Node

This section describes the process for installing multiple instances of GSS in a stand-alone deployment.

### Using the Installation Script - Stand-alone Node


Installation is accomplished using the `inst_serv` script to install GSS binaries. It provides a menu-driven interface with question prompts. Most prompts display default values or information derived from the server's current setup - such as IP addresses for configured interfaces.

The following information will help you use the installation script most effectively:

- **Ctrl-C** will abort the installation process at any time during the procedure.
- To accept defaults and continue the process, enter **n** as your choice for any prompt.
- To change a displayed value, enter the menu item number to display an entry field or a list of possible values.

The information from the prompts is used to generate the GSS configuration file (`gss.cfg`). This file can be changed at anytime after the installation.

---

 **Important:** It is recommended that you fill in path prompts only after you have created the directories to be used.

---

## Installation Procedure - Stand-alone Node


The following procedure assumes that you are logged in to the GSS server with root privileges and that you are starting at the root directory level.

**Step 1** Change to the `/<packages>/gss_<version>` directory where you stored the GSS application software in *step 5* of the previous section.

**Step 2** Locate the installation script file `inst_serv` and execute the following command:

```
./inst_serv
```

---

 **Important:** This script will check the version of the operating system installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Stand-alone Deployment* section, the script will abort the GSS installation.

---

The following appears, with pauses for validation, after entering the `inst_serv` command:

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
required changes in '/etc/system' file, restart the machine to get these
changes reflected and then start installation again.
```

```
set msgsys:msginfo_msgmnb=65536
```

```
set msgsys:msginfo_msgtql=1024
```

```
set shmsys:shminfo_shmmax=33554432
```

```
set shmsys:shminfo_shmmin=1
```

```
set shmsys:shminfo_shmmni=256
```

```
set shmsys:shminfo_shmseg=256
```

```
set semsys:seminfo_semmap=256
```

```
set semsys:seminfo_semmni=512

set semsys:seminfo_semmns=512

set semsys:seminfo_semmsl=32
```

Standalone Mode Installation [in standalone environment]...

**Step 3** If you did not make the changes to the system file, then abort now (**CTRL-C**), make the changes to the system file, and then reboot. After rebooting, begin the installation procedure again.

If you made changes to the system file as in the *Installation First Steps* section, then continue to the next step.

```
GTPP Storage Server installation directory
```

```
Path where Gss will be installed :/opt
```

```
GSS Installation dir [/opt] ? </home/export/install_8_0_xx>
```

**Step 4** Enter the name of the directory where the GSS active components are to be installed, for example `<install_dir>`. It is recommended that you include this directory at the root level. The installation script creates the directory if needed.

Shortly after typing `<install_dir>` and pressing **Enter**, the following appears:

```
Entering n will save configuration values and take you to next
configuration, To change the default values, enter option number
```

```
*** PostgreSQL installation configuration ***
```

```
1) PostgreSQL port : 5432
```

```
2) PostgreSQL login : postgres
```

```
3) PostgreSQL passwd : postgres
```

```
n) Proceed to next configuration
```

```
a) Abort Installation
```

```
Enter Your Choice : [n] ?
```

```
Entering n will save configuration values and take you to next
configuration, To change the default values, enter option number
```



**Important:** All values that appear initially for this menu are system defaults and you do not need to make changes if the values are acceptable.

**Step 5** Enter the line number to change a parameter value, if needed. Then enter **n** to save changes (if made) or defaults and move to the next menu.

```
*** GSS
```

```
Configuration Parameters ***
```

```
1) File Format for data files : starent
```

```
2) Hard Limit Interval for File Generation (mins) : 0
```

- 3) Support for LRSN rewrite : n
- 4) Encoding of IP Address in binary format : n
- 5) Enable redundant data file support : n
- 6) GSN Location : GSN
- p) Go back to previous menu
- n) Proceed to next configuration
- a) Abort Installation



**Important:** The GSS and FileGen are set to run in archive mode by default. The archive mode, used in deployments that do not include CGFs, instructs the server to save records to a file.

**Step 6** Enter a line number to change a parameter value, if needed. This can be repeated as often as necessary to change parameter values. Then enter **n** to save changes (if made) or defaults and move to the next menu.

```

*** Network Interface Configurations ***

Currently configured IP interfaces on the machine : 10.8.1.205

1 ) 10.1.1.111
2 ) 123.1.2.33

p ) Go back to previous menu
n ) Proceed to next configuration
a ) Abort Installation

Enter your choice : n

```



**Important:** Note that the script has sensed both the number of interfaces and their IP addresses.

**Step 7** Enter a line number to change a parameter value, if needed. This can be repeated as often as necessary to change parameter values. Then enter **n** to save changes (if made) or defaults.

Pressing **n** completes the menu-driven portion of the installation process and displays the configuration that you have created.

```

=====
Standalone Mode Installation
=====

GSS installation path : /<install_dir>/gss

*** PostgreSQL Configurations ***

```

```

PostgreSQL port : 5432
PostgreSQL login : postgres
PostgreSQL passwd : postgres
**** GSS Configurations ****
File Format for data files : starent
Hard Limit Interval for File Generation : 0
GTPP Dictionary : custom1
Support for LRSN rewrite : n []
Encoding of IP Address in binary format : n
GSN Location : GSN
Enable redundant data file support : n
**** Network Host Configurations ****
IP Address of the machine to be used : 10.1.1.111
=====

```

You are given the opportunity to modify the configuration that you have created. (The values displayed above were entered for illustration and not as recommendations.)

```
Do you want to Modify Configuration [n] ? n
```

**Step 8** Press **y** to return to the menus and change the configuration or press **n** to continue the installation process.

```

Installing GSS..... Please wait.....
Extracting perl tar... Done.

Add following entry to crontab (if not already present) to remove
processed data files in /<install_dir>/gss/data after storage period of 7
day(s)

0 * * * * /<install_dir>/gss/bin/cleanup.sh >>
/<install_dir>/gss/log/cleanup.log 2>&1

To start Process Monitor Tool along with GSS and Filegen (if not started
already) : execute "/<install_dir>/gss/serv start"

To get help on "GSS": execute "/<install_dir>/gss/serv help"

For additional info and performance tuning please read README & GSS User
Guide in doc directory

Do You Want To Start GSS : [y] ? y

```

This is the last action that you must take to complete the installation process.

**Step 9** Enter **n** to complete the installation without starting the GSS. Press return to accept the **yes** default to complete the installation process and start the GSS.

```
This will start Process Monitor Tool along with GSS and Filegen using
params listed in /<install_dir>/gss/etc/gss.cfg
```

```
Please see log/psmon.log file for log messages
```

```
Starting Process Monitor Tool...
```

```
Done.
```

```
Capturing status, please wait for a while...
```

```
=====
0 1118 12:04:16 TS 59 0:00 <install_dir>/gss/bin/gssfilegen 1
0 1015 12:04:10 TS 59 0:00 /usr/bin/bash <install_dir>/gss/serv start
29661
0 1102 12:04:15 TS 59 0:00 <install_dir>/gss/lib/perl5.8.5/bin/perl
<install_dir>/gss/psmon --daemon --cro 1
0 1113 12:04:16 TS 59 0:00 <install_dir>/gss/bin/gss 1
=====
GTPP Storage Server Version 8.0.xx installation done.
```

The status display indicates that GSS, FileGen, PSMON, and PostgreSQL have all been started. If nothing displays, turn to the *Troubleshooting the GSS* section in the *GTPP Storage Server Administration* chapter. In most cases, if the other components are started, then the PostgreSQL has also been started.

**Step 10** Optional: Confirm the location of the installed GSS components.

```
# cd /<install_dir>

# ls -al

total 16

drwxr-xr-x 4 root other 512 Dec 9 17:29 .
drwxr-xr-x 11 root root 512 Dec 9 17:28 ..
-rwxrwxrwx 1 root other 3782 Dec 9 17:28 StandaloneGSSUninstall.sh
drwxrwxr-x 12 root root 512 Dec 9 17:29 gss

drwxr-xr-x 6 postgres other 512 Dec 9 17:28 postgres
```

**Step 11** Optional: Check the location of the GSS application files, tools and engineering documentation.

```
# cd gss
# ls -al
total 232
drwxrwxr-x 12 root root 512 Dec 21 22:43 .
drwxr-xr-x 4 root other 512 Dec 21 22:43 ..
-rw-r--r-- 1 postgres other 533 Aug 17 10:58 .configfile
-r--r--r-- 1 postgres root 1168 Aug 17 10:58 .gss.env
-rw----- 1 root other 5 Dec 21 22:43 .gss.pid
-rw----- 1 root other 5 Dec 21 22:43 .gssfilegen.pid
-rw----- 1 root other 7 Dec 21 22:43 .gssfilegen.seq
-rw-r--r-- 1 root other 0 Dec 21 22:42 .inst_serv.err
-rwxr-xr-x 1 root root 4057 Mar 3 2006 README
drwxrwxr-x 2 root root 512 Dec 21 22:42 bin
drwxrwxr-x 2 root root 512 Sep 8 2004 data
drwxrwxr-x 2 root root 512 May 31 2005 doc
drwxrwxr-x 2 root root 512 Dec 21 22:42 etc
-rwxr-xr-x 1 root other 22480 Dec 21 22:42 gss_ctl
drwxrwxr-x 3 root root 512 Dec 21 22:42 lib
drwxrwxr-x 3 root root 512 Dec 21 22:43 log
-rwxr-xr-x 1 root other 54445 Dec 21 22:42 psmon
-rw-r--r-- 1 root other 4 Dec 21 22:43 psmon.pid
-rwxr-xr-x 1 root other 19850 Dec 21 22:42 serv
drwxrwxr-x 2 root root 512 Oct 10 05:43 sql
drwxrwxr-x 2 root root 512 Oct 10 05:43 template
drwxrwxr-x 2 root root 512 Sep 8 2004 tmp
drwxrwxr-x 3 root root 512 Dec 21 22:42 tools
```

**Step 12** Optional: Locate the GSS configuration file (*gss.cfg*).

```
# cd etc
# ls -al
```

```
total 42
drwxrwxr-x 2 root root 512 Dec 21 22:42 .
drwxrwxr-x 12 root root 512 Dec 21 22:43 ..
-rw-r--r-- 1 root other 10921 Dec 21 22:42 gss.cfg
-rw-r--r-- 1 root other 2459 Dec 21 22:42 gsslogger.xml
-rw-r--r-- 1 root other 3690 Dec 21 22:42 psmon.cfg
-rw-r--r-- 1 root other 261 Dec 21 22:42 uninstall_config_file
```

Before working with the GSS, it is recommended to create a write-protected copy of the *gss.cfg* file and store it in a separate directory. To ensure you remember the configuration for your software version, we suggest that you store the in the */<packages>/<gss\_version>* directory.

## Installing multiple GSS Instances - Node 1 in Cluster

This section describes the process for installing multiple GSS instances on the primary GSS node of the cluster.

Prior to installing the GSS application, ensure that the cluster is installed and configured as needed. For information on installing and configuring the Sun cluster, refer to the *Sun documentation*.

### Using the Installation Script - Cluster Node 1

Installation is accomplished using the **inst\_serv** script. It provides a menu-driven interface with question prompts. Most prompts display default values or information derived from the server's current setup.

The following information will help you use the installation script most effectively:

- **Ctrl-C** will abort the installation process at any time during the procedure.
- To accept defaults and continue the process, enter **n** as your choice for any prompt.
- To change a displayed value, enter the menu item number to display an entry field or a list of possible values.

The information from the prompts is used to generate the GSS configuration file (**gss.cfg**). Using a text editor, this file can be changed at anytime after the installation.

### Installation Procedure - Node 1

The following procedure assumes that you are logged in to the GSS server with root privileges and that you are starting from the root directory.

**Step 1** Change to the */<packages>/gss\_<version>* directory where you stored the GSS application software.

**Step 2** Locate the installation script file **inst\_serv** and execute the following command:

`./inst_serv`



**Important:** This script will check the version of operating system and cluster software installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Cluster Deployment* section, the script will abort the GSS installation.

The following appears, with pauses for validation, after entering the `inst_serv` command.

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
required changes in '/etc/system' file, restart the machine to get these
changes reflected and then start installation again.
```

```
set msgsys:msginfo_msgmb=65536
```

```
set msgsys:msginfo_msgtql=1024
```

```
set shmsys:shminfo_shmmax=33554432
```

```
set shmsys:shminfo_shmmin=1
```

```
set shmsys:shminfo_shmmni=256
```

```
set shmsys:shminfo_shmseg=256
```

```
set semsys:seminfo_semmap=256
```

```
set semsys:seminfo_semmni=512
```

```
set semsys:seminfo_semmns=512
```

```
set semsys:seminfo_semmsl=32
```

```
Using cluster for Installation
```

```
Cluster Mode Installation (To be used in cluster environment) [n] ? y
```



**Important:** Note that the script senses whether the server is a Stand-alone node or a Cluster node. If you made changes to the system file as in the [Installation First Steps](#) section, then you can continue. If not,

abort the installation using **CTRL+C**, make the changes to the system file, and then reboot. After rebooting, begin the installation procedure again.

---

**Step 3** Enter **y (yes)** to continue the installation.

```
GTPP Storage Server installation directory


Specify the common path for GSS data and logs on all cluster nodes
[/sharedgss/gss] ?
```

**Step 4** Press **Enter** to accept the default directory `/sharedgss/gss` or enter the name of another directory. Next you are prompted for the location to install the GSS.

```
Path where Gss will be installed : /gss1

GSS Installation dir [/gss1] ? /clustserver/gss1
```

---

 **Important:** In case of cluster mode, it is highly recommended that you do not install the GSS application in the `/opt`, `/opt/gss/`, or `/opt/postgres` directory.


---

**Step 5** Press **Enter** to accept the default or enter the name of the directory where the GSS active components are to be installed. It is recommended that you put this directory at the root level. The installation script creates the directory if needed.

Shortly after responding to the prompt for the installation directory, the following appears:

```
Do you want Backup installation for current cluster mode installation
[y/n]: [n] ?
```

---

 **Important:** This enables backup mode for the GSS node in a cluster deployment.

---

If you do not want the backup for cluster mode installation, proceed to *step 7*. Otherwise, continue with next step.

**Step 6** Enter **y (yes)** to enable node switchover. The installation continues with a menu to configure the PostgreSQL parameters for backup.

```
*** PostgreSQL configuration for backup Installation ***

1) PostgreSQL port for backup installation :

2) PostgreSQL login for backup installation :

3) PostgreSQL data directory for backup installation :

n) Proceed to next configuration

a) Abort Installation

Enter Your Choice : [n] ?
```

**Step 7** Enter the number or letter of your choice and enter the postgres details as needed. The installation continues with a menu to configure the PostgreSQL parameters.

Entering n will save configuration values and take you to next configuration. To change the default values, enter option number

```
*** PostgreSQL installation configuration ***

1) PostgreSQL port : 5432
2) PostgreSQL login : postgres
3) PostgreSQL passwd : postgres
4) Shared PostgreSQL dir : /sharedpostgres/gss1
4) Shared PostgreSQL User ID :
n) Proceed to next configuration
a) Abort Installation


Enter Your Choice : [n] ?
```

**Step 8** Enter the number or letter of your choice. Make changes as needed.

Entering n will save configuration values and take you to next configuration. To change the default values, enter option number

**Step 9** Enter **n** to save changes or defaults and move to the next menu.

---

 **Important:** The GSS and FileGen are set to run in archive mode by default. The archive mode, used in deployments that do not include CGFs, instructs the server to save records to a file.

---

```
*** GSS Configuration Parameters ***

1) File Format for data files : starent
2) Hard Limit Interval for File Generation (mins) : 0
3) Support for LRSN rewrite : n
4) Encoding of IP Address in binary format : n
5) GSS Port : 50000
6) Enable redundant data file support : n
7) GSN Location : GSN
8) Specify the GTPP Dictionary : custom6
p) Go back to previous menu
n) Proceed to next configuration
a) Abort Installation
```



**Important:** The **Specify the GTPP Dictionary** option appears only if the **Support for LRSN rewrite** or **Encoding of IP Address in binary format** parameter is enabled. Otherwise, the GTPP dictionary is set to default.

**Step 10** Enter the number or letter of your choice. Make changes as needed. Then enter **n** to save changes or defaults and move to the next menu.

```
*** Network Interface Configuration ***

1) Logical Host IP Addresss :
2) Logical Host Name :
3) Additional Logical Host Name[eg.For Mediation Server] : n
p) Go back to previous menu
n) Proceed to next configuration
a) Abort Installation

Enter your choice : n
```

**Step 11** Enter option 1.

```
Enter Your Choice :
1

Please
specify already available logical host Address for GSS cluster :
?
```

**Step 12** Enter the IP address of the logical host, press **Enter**, and move to the next prompt.

```
Enter Your Choice :
```

**Step 13** Enter option 2.

```
Please specify Logical hostname for above logical host address : ?
```

**Step 14** Enter a line number to change any parameter value, if needed. This can be repeated as often as necessary to change parameter values. Then enter **n** to save changes (if made) or defaults.

Pressing **n** completes the menu-driven portion of the installation process and displays the configuration that you have created.

```
=====
Cluster Mode Installation
=====

GSS installation path : /clustserver/gss1/gss
Common path for GSS data and logs : /sharedgss/gss1
```

```

*** Backup PostgreSQL Configurations ***

PostgreSQL port for backup installation : 5477

PostgreSQL login for backup installation : backpost

PostgreSQL data directory for backup installation : /backpost

*** PostgreSQL Configurations ***

PostgreSQL port : 5432

PostgreSQL login : gsspg

PostgreSQL passwd : gsspg

Shared PostgreSQL dir : /sharedpostgres

Shared PostgreSQL UID : 10006

**** GSS Configurations ****

File Format for data files : custom7

Hard Limit Interval for File Generation : 2

GTPP Dictionary : custom6

Support for LRSN rewrite : n []

Encoding of IP Address in binary format : n

GSN Location : GSN

GSS Port : 50000

Enable redundant data file support : n

*** Network Host Configurations ***

Logical Host IP Addresss : 10.1.1.1

Logical Host Name : gssserv

Additional Logical Host Name : n

=====

```

You are given the opportunity to modify the configuration that you have created.

Do you want to Modify Configuration [y/n] : [n] ?

**Step 15** Press **y** to return to the menus and change the configuration or press **n** or **Enter** to continue the installation process.

Installing GSS..... Please wait.....

Extracting perl tar...

Done.

Add following entry to crontab (if not already present) to remove processed data files in /sharedgss/gss/data after storage period of 7 day(s)

```
0 * * * * /TEST_GSS/cvserver/gss/bin/cleanup.sh >>
/sharedgss/gss/clustems1_log//cleanup.log 2>&1
```

For additional info and performance tuning please read README & GSS User Guide in doc directory

GSS Cluster Agent Installation configuration

Proceeding with GSS Cluster Agent Installation

Extracting StarentGss.tar...

Processing package instance <StarentGss> from </TEST\_GSS/gss\_x\_x\_xx>

Sun Cluster resource type for Gss server(sparc) 3.0.0,REV=

Sun Microsystems, Inc.

Using </opt> as the package base directory.

## Processing package information.

## Processing system information.

## Verifying package dependencies.

## Verifying disk space requirements.

## Checking for conflicts with packages already installed.

## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user permission during the process of installing this package.

Do you want to continue with the installation of <StarentGss> [y,n,?] y

**Step 16** Enter y (yes) to continue the installation.

Installing Sun Cluster resource type for Gss server as <StarentGss>

## Installing part 1 of 1.

/opt/StarentGss/README.Gss

/opt/StarentGss/bin/Gss\_mon\_check.ksh

/opt/StarentGss/bin/Gss\_mon\_start.ksh

/opt/StarentGss/bin/Gss\_mon\_stop.ksh

```

/opt/StarentGss/bin/Gss_probe.ksh
/opt/StarentGss/bin/Gss_svc_start.ksh
/opt/StarentGss/bin/Gss_svc_stop.ksh
/opt/StarentGss/bin/Gss_update.ksh
/opt/StarentGss/bin/Gss_validate.ksh
/opt/StarentGss/bin/gethostnames
/opt/StarentGss/bin/gettime
/opt/StarentGss/bin/hasp_check
/opt/StarentGss/bin/simple_probe
/opt/StarentGss/etc/Starent.Gss
/opt/StarentGss/man/man1m/Gss_config.1m
/opt/StarentGss/man/man1m/removeGss.1m
/opt/StarentGss/man/man1m/startGss.1m
/opt/StarentGss/man/man1m/stopGss.1m
/opt/StarentGss/util/Gss_config
/opt/StarentGss/util/removeGss
/opt/StarentGss/util/startGss
/opt/StarentGss/util/stopGss
[ verifying class <none> ]
## Executing postinstall script.
Installation of <StarentGss> was successful.
GTPP Storage Server Version 9.0.106 Instance Creation done.
GSS installation on node1 is complete. please proceed with installation
on node2....

Do you want to add another GSS instance [n] ?

```

**Step 17** Enter **y** to create another GSS instance. Repeat the above steps in order to add and modify the new GSS instance with configurations as required.

```

Do you want to add another GSS instance [n] ? y

```

---

```

Instance Name : GSS1

```

```
Instance Path : /clustserver/GSS1/gss
```

```
Instance Status : Running
```

---

```
Instance Name : GSS2
```

```
Instance Path : /clustserver/GSS2/gss
```

```
Instance Status : Running
```

---

The status display indicates the name, path and status of the instance.

## Installing multiple GSS Instances - Node 2 in Cluster

This section describes the process for installing multiple GSS instances on to a second GSS node in a cluster setup.


The following procedure assumes that you are logged in to the node 2 (standby) GSS server with root privileges and that you are starting from the root directory.

**Step 1** Change to the `/gss_<version>` directory where the GSS files are stored.

**Step 2** Locate the installation script file `inst_serv` and execute the following command:

```
./inst_serv
```

---

 **Important:** This script will check the version of operating system and cluster software installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Cluster Deployment* section, the script will abort the GSS installation.

---

The following appears, with pauses for validation, after entering the `inst_serv` command.

```

S T A R E N T G T P P S T O R A G E S E R V E R

Copyright(c) 2009 Starent Networks Corp. All rights reserved.

===== Mon Dec 14 17:03:00 IST 2009 =====

Checking For Root Privileges ..... Done

Start installing new version of GSS: [y] ?

Extracting GSS package ..... Done

Extracting Perl ..... Done.

GSS Cluster Agent Installation ..... Done

=====

```

```
GSS basic package installed at /opt/gss_global.
```

```
For creating multiple GSS instances, please run
/opt/gss_global/make_gss_instance
```

```
=====
```

**Step 3** Run `/opt/gss_global/make_gss_instance` to create a GSS instance.

```
bash-3.00# /opt/gss_global/make_gss_instance
```

```
Cluster Mode Installation (To be used in cluster environment) [y/n]: [n]
? y
```

**Step 4** Enter **y (yes)** to continue the installation.

```
Using cluster for Installation
```

```
Specify the common path for GSS data and logs on all cluster nodes
[/sharedgss/gss1] ?
```

```
GTPP Storage Server installation directory
```

**Step 5** Enter the name of the directory where files that will be common to all nodes in the cluster are to be stored, for example `<shared_gss_dir>`. You must enter the same path when you installed the GSS on node 1.

Shortly after typing `<shared_gss_dir>` and pressing **Enter**, the configuration created for cluster node 1 appears.

```
=====
```

```
Cluster Mode Installation
```

```
=====
```

```
GSS installation path : /clustserver/GSS1/gss
```

```
Common path for GSS data and logs : /sharedgss/gss1
```

```
GSS Instance Name : GSS1
```

```
*** PostgreSQL Configurations ***
```

```
PostgreSQL port : 5432
```

```
PostgreSQL login : 99999
```

```
PostgreSQL passwd : postgres
```

```
Shared PostgreSQL dir : /sharedpostgres/GSS1/postgres/data
```

```
Shared PostgreSQL UID : 99999
```

```
**** GSS Configurations ***
```

```
Archive Mode : y
```

```
File Format for data files : custom3
```

```
Hard Limit Interval for File Generation : 15
```

```
GTPP Dictionary : custom3
```

```
Support for LRSN rewrite : n []
```

```
Encoding of IP Address in binary format : n
```

```
GSN Location : GSN
```

```
GSS Port : 50000
```

```
Enable redundant data file support : n
```

```
*** Network Host Configurations ***
```

```
Logical Host IP Addresss : 192.168.143.100
```

```
Logical Host Name : gssserv
```

```
Additional Logical Host Name : n
```

```
=====
```

```
If you want to change configuration values, do not start GSS after
installation. Change required configuration parameters from
/GSS1/gss/etc/gss.cfg and then start GSS..
```

```
Proceed With Installation [y/n] : [y] ?
```

**Step 6** Press **Enter** to continue the installation.

```
Installing GSS..... Please wait.....
```

```
Add following entry to crontab (if not already present) to remove
processed data files in /sharedgss/gss1/data after storage period of 7
day(s)
```

```
0 * * * * /clustserver/GSS1/gss/bin/cleanup.sh >>
/clustserver/GSS1/gss/log/cleanup.log 2>&1
```

```
Starting second node installation.....
```

```
Do you want to start GSS on cluster nodes [y/n] : [y] ?
```

```
Starting GSS on cluster node....
```

```
Capturing status, please wait for a while...
```

```
=====
```

```
0 21170 17:04:12 TS 59 00:00 /clustserver/GSS1/gss/bin/gssfilegen 1
```

```
0 21154 17:04:11 TS 59 00:01
/clustserver/GSS1/gss/lib/perl5.8.5/bin/perl /GSS1/gss/psmon --daemon --
cron /clustserver/GSS1/gss 1
```

```
0 21165 17:04:12 TS 59 00:00 /clustserver/GSS1/gss/bin/gss 1

99999 21187 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21189 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21115 1 0 17:04:07 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21184 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21173 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21177 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21122 21115 0 17:04:07 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21175 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21188 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21191 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21182 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21179 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21174 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data
```

```
99999 21190 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21183 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21185 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21123 21115 0 17:04:07 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21186 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21176 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21181 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21117 21115 0 17:04:07 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21178 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21180 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21195 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21194 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data

99999 21197 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data
```

```
99999 21192 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data
```

```
99999 21196 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data
```

```
99999 21193 21115 0 17:04:13 ? 0:00
/opt/gss_global/global/postgres/bin/postgres -D
/clustserver/GSS1/postgres/data
```

```
=====
```

GTPP Storage Server Version 9.0.106 Instance Creation done.

- Step 7** Enter **y** to create another GSS instance. Repeat the above steps in order to add and modify the new GSS instance with configurations as required.

```
Do you want to add another GSS instance [n] ? y
```

```
-----
Instance Name : GSS1
```

```
Instance Path : /clustserver/GSS1/gss
```

```
Instance Status : Running
```

```
-----
Instance Name : GSS2
```

```
Instance Path : /clustserver/GSS2/gss
```

```
Instance Status : Running
-----
```

The status display indicates the name, path and status of the instance.

- Step 8** Optional. Return to the command line of the primary node. Confirm that the PostgreSQL engine has started by entering the following command:

```
ps -ef
```

The resulting display will display all the processes running and there will most likely be multiple lines that confirm the PostgreSQL is running as they look similar to the following:

```
postgres 15080 14972 0 07:23:25 ? 0:00
/<clus_install_dir>/postgres/bin/postgres -D
```

- Step 9** Optional. Locate the GSS configuration file (*gss.cfg*).

```
# cd etc
```

```
# ls -al
```

```
total 38
drwxr-xr-x 2 1460 100 512 Apr 20 07:22 .
drwxr-xr-x 12 1460 100 512 Apr 20 07:23 ..
-rw-r--r-- 1 root other 12561 Apr 20 07:22 gss.cfg
-rw-r--r-- 1 root other 3521 Apr 20 07:22 psmon.cfg
```

Before doing anything with the GSS, it is recommended to create a write-protected copy of the `gss.cfg` file and store it in a separate directory. To ensure you remember the configuration for your software version, we suggest that you store it in the `/<packages>/<gss_version>` directory.

## Uninstalling Multiple GSS Instances - Stand-alone Node

This section describes the process for uninstalling multiple instances of GSS and all of the associated GSS components, for a stand-alone deployment.

### Using the Uninstallation Script

Uninstallation is performed using the `uninstall` script that needs to be run for each instance of GSS and then for global instance on standalone and cluster modes. This removes the shared directories and installation directories. It is recommended to ensure the following:

- Place the tar ball and all its contents in the base directory.
- Run the `uninstall` script from the respective base directory.
- Copy and rename the old `gss.cfg` file to a safe directory.

### Uninstallation Procedure - Stand-alone Node

The `StandaloneGSSUninstall.sh` uninstallation script is the script used for uninstall process on a standalone mode. This script created in each GSS instance installation directory. For example, GSS instance installation directory is `/cvserver` then, uninstallation script is created in `/cvserver` as `StandaloneGSSUninstall.sh`.

Uninstalling the respective instance of GSS in a standalone mode accomplishes the following:

- Deletes `gss` directory from GSS installation path
- Deletes `postgres` directory from `postgres` installation path
- Deletes the `postgres` user created
- Cleans shared memory and semaphores for the given `postgres` user
- Removes `postgres` lock files from `/tmp` for given `postgres` port

**Step 1** Change to the directory where the first instance of GSS is installed.

```
# cd /<instance_dir>
```

**Step 2** Locate the uninstall script and run the script to begin the uninstall process for the first instance installed.

```
Uninstallation of instance 1
```

```
bash-3.00# cd GSS1
```

```
bash-3.00# ls
```

```
StandaloneGSSUninstall.sh gss postgres#
```

```
bash-3.00# ./StandaloneGSSUninstall.sh
```

```
/GSS1/gss
```

```
Uninstallation Process will remove shared directories, installation
directories....
```

```
Do You Want To Proceed With Uninstallation [y/n] : [n] ? y
```

**Step 3** Enter **y (yes)** to continue the uninstall process.

The system goes to the installation directory where the system-generated *uninstall\_config\_file* resides and uses the information in that file to complete the install process.

```
This will stop Process Monitor Tool along with GSS and Filegen
```

```
Please see log/psmon.log file for log messages
```

```
Stopping Process Monitor Tool...
```

```
Done.
```

```
Stopping GSS...
```

```
Done.
```

```
Stopping GSS FileGen...
```

```
Done.
```

```
Partial FileGen not running
```

```
Final FileGen not running
```

```
waiting for server to shut down.... done
```

```
server stopped
```

```
starting cleanup process
```

```
Cleaning the gss installation paths ...
```

```
Cleaning the postgres installation paths ...
```

```
Cleaning the shared memory and semaphores ...
```

```

Deleting the postgres user created during installation ...
Cleaning the postgres lock from /tmp directory ...
Uninstallation process completed successfully.....

```

The first instance is uninstalled.

**Step 4** To unistall the second instance, change to the directory where the second GSS instance is installed.

```

Uninstallation of instance 2
bash-3.00# cd GSS2/
bash-3.00# ls
StandaloneGSSUninstall.sh gss postgres
bash-3.00# ./StandaloneGSSUninstall.sh
/GSS2/gss
Uninstallation Process will Remove Installation directories
Do You Want To Proceed With Uninstallation [y/n]: [n] ? y

```

**Step 5** Enter **y (yes)** to continue the uninstall process.

```

This will stop Process Monitor Tool along with GSS and Filegen
Please see log/psmon.log file for log messages
Stopping Process Monitor Tool...
Done.
Stopping GSS...
Done.
Stopping GSS FileGen...
Done.
Partial FileGen not running
Final FileGen not running
waiting for server to shut down.... done
server stopped
starting cleanup process
Cleaning the gss installation paths ...
Cleaning the postgres installation paths ...

```

```

Cleaning the shared memory and semaphores ...
Deleting the postgres user created during installation ...
Cleaning the postgres lock from /tmp directory ...
Uninstallation process completed successfully.....

```

- Step 6** Uninstall the global instance after removing each instance of GSS. Change to the `/opt/gss_global/` directory where the `Global_Gss_Uninstall.sh` is placed. This script deletes the installation directories from `/opt/gss_global` directory and removes the `starentGss` package.

```

bash-3.00# cd /opt/gss_global/
bash-3.00# ls
Global_Gss_Uninstall.sh bin make_gss_instance sc_event.dtd
Logs global nvpair.dtd sc_reply.dtd
bash-3.00# ./Global_Gss_Uninstall.sh
Uninstallation Process will remove shared directories, installation
directories....

Do You Want To Proceed With Uninstallation [y/n]: [n] ? y

```

- Step 7** Enter **y (yes)** to continue the uninstall process.

```

Starting cleanup process
Cleaning the starentGss package ...
The following package is currently installed:
StarentGss Sun Cluster resource type for Gss server (sparc) 3.0.0,REV=
Do you want to remove this package? [y,n,?,q] y

```

- Step 8** Enter **y (yes)** to remove the installed package instance.

```

## Removing installed package instance <StarentGss>

This package contains scripts which will be executed with super-user
permission during the process of removing this package.

Do you want to continue with the removal of this package [y,n,?,q] y

```

- Step 9** Enter **y (yes)** to continue the uninstall process.

```

## Verifying package <StarentGss> dependencies in global zone
## Processing package information.
## Executing preremove script.

```

```
Resource <Gss-hars> has been removed already
Resource type <Starent.Gss> has been un-registered already
Network Resource not removed...
You may run removeGss again with the -h option to remove network
resource.

## Removing pathnames in class <none>

/opt/StarentGss/util/stopGss
/opt/StarentGss/util/startGss
/opt/StarentGss/util/removeGss
/opt/StarentGss/util/Gss_config
/opt/StarentGss/util
/opt/StarentGss/man/man1m/stopGss.1m
/opt/StarentGss/man/man1m/startGss.1m
/opt/StarentGss/man/man1m/removeGss.1m
/opt/StarentGss/man/man1m/Gss_config.1m
/opt/StarentGss/man/man1m
/opt/StarentGss/man
/opt/StarentGss/etc/Starent.Gss
/opt/StarentGss/etc
/opt/StarentGss/bin/simple_probe
/opt/StarentGss/bin/hasp_check
/opt/StarentGss/bin/gettime
/opt/StarentGss/bin/gethostnames
/opt/StarentGss/bin/Gss_validate.ksh
/opt/StarentGss/bin/Gss_update.ksh
/opt/StarentGss/bin/Gss_svc_stop.ksh
/opt/StarentGss/bin/Gss_svc_start.ksh
/opt/StarentGss/bin/Gss_probe.ksh
/opt/StarentGss/bin/Gss_mon_stop.ksh
```

```
/opt/StarentGss/bin/Gss_mon_start.ksh
/opt/StarentGss/bin/Gss_mon_check.ksh
/opt/StarentGss/bin
/opt/StarentGss/README.Gss
/opt/StarentGss
## Executing postremove script.
## Updating system information.
Removal of <StarentGss> was successful.
Global GSS uninstalled ...
bash-3.00#
```

## Uninstalling multiple GSS Instances - Cluster Nodes

This section describes the process for uninstalling multiple instances of GSS and all of the associated GSS components, for a cluster deployment.

### Using the Uninstallation Script

Uninstallation is performed using the uninstall script that needs to be run for each instance of GSS and then for global instance on standalone and cluster modes. This removes the shared directories and installation directories. It is recommended to ensure the following:

- Place the tar ball and all its contents in the base directory.
- Run the uninstall script from the respective base directory.
- Copy and rename the old **gss.cfg** file to a safe directory.

### Uninstallation Procedure - Cluster Node

The **ClusterGSSUninstall.sh** uninstallation script is the script used for uninstall process on a standalone mode. This script created in each GSS instance installation directory. For example, GSS instance installation directory is */cvserver* then, uninstallation script is created in */cvserver* as *ClusterGSSUninstall.sh*. During both the installation and upgrade processes, the system generates system-specific uninstall files: an uninstall script and an uninstall configuration file.

Uninstalling the respective instance of GSS in a cluster mode accomplishes the following:

- Brings resource group offline from both the nodes.

- Disables CRNP, GSS resource, network resource and additional resource (if defined).
- Unmanages resource group.
- Deletes CRNP, gss, network resource and additional resource.
- Unregisters CRNP resource type, GSS resource type and finally removes the resource group.
- Deletes gss directory from GSS shared data path and postgres directory from postgres shared data path.
- Deletes gss directory from GSS installation path.
- Deletes postgres directory from postgres installation path.
- Cleans shared memory and semaphores for the given postgres user.
- Deletes the postgres user created.
- Removes postgres lock files from /tmp for given postgres port.

**Step 1** Change to the directory where the first instance of GSS is installed. Begin with the primary node 1.

```
# cd /<instance_dir>
```

**Step 2** Run the uninstall script to begin the uninstall process.

```
Uninstallation of instance 1 from node 1
bash-3.00# cd GSS1
bash-3.00# ls
ClusterGSSUninstall.sh gss postgres
bash-3.00# ./ClusterGSSUninstall.sh
Uninstallation Process will remove shared directories, installation
directories....
Do You Want To Proceed With Uninstallation [y/n] : [n] ? y
```

**Step 3** Enter **y (yes)** to continue the uninstall and start the cleanup operation.

```
starting cleanup process
checking if resource group exists..
resourcegroup GSS1 exists
Bringing Resource Group offline from both the nodes.....
removing cluster resources..
Bringing the resource group offline..
Disabling Gss-hars_GSS1
Disable gssserv
Disable GSS1_crnp
```

```

unmanaging GSS1
removing CRNP
invalid resource
clresource: (C615326) Will not attempt to delete resource "CRNP".
removing Gss-hars_GSS1
removing gssserv
clresource: (C383672) Will not attempt to delete "gssserv" as another
resource in the system has dependency on it.
removing GSS1_crnp
Remove the resource group
Cleaning the gss and postgres common data paths ...
Cleaning the gss installation paths
Cleaning the postgres installation paths ...
Cleaning the /opt directories paths ...
Cleaning the shared memory and semaphores
Deleting the postgres user created during installation ...
Cleaning the postgres lock from /tmp directory
Uninstallation process completed successfully.....

```

**Step 4** Continue to unistall the second instance. Change to the directory where the second GSS instance is installed.

```

Uninstallation of instance 2 from node 1
bash-3.00# cd GSS2
bash-3.00# ls
ClusterGSSUninstall.sh gss postgres
bash-3.00# ./ClusterGSSUninstall.sh
Uninstallation Process will remove shared directories, installation
directories....
Do You Want To Proceed With Uninstallation [y/n] : [n] ? y

```

**Step 5** Enter y (yes) to continue the uninstall process.

```

starting cleanup process
checking if resource group exists..

```

```

resourcegroup GSS2 exists
Bringing Resource Group offline from both the nodes.....
removing cluster resources..
Bringing the resource group offline..
Disabling Gss-hars_GSS2
Disable gssserver
Disable GSS2_crnp
unmanaging GSS2
removing CRNP
invalid resource
clresource: (C615326) Will not attempt to delete resource "CRNP".
removing Gss-hars_GSS2
removing gssserver
clresource: (C383672) Will not attempt to delete "gssserver" as another
resource in the system has dependency on it.
removing GSS2_crnp
Remove the resource group
Cleaning the gss and postgres common data paths ...
Cleaning the gss installation paths
Cleaning the postgres installation paths ...
Cleaning the /opt directories paths ...
Cleaning the shared memory and semaphores
Deleting the postgres user created during installation ...
Cleaning the postgres lock from /tmp directory
Uninstallation process completed successfully.....

```

**Step 6** The next step is to uninstall the global instance from node 1. Change to the */opt/gss\_global/* directory.

```

Uninstallation of Global instance from node 1
bash-3.00# cd /opt/gss_global/
bash-3.00# ls

```

```
Global_Gss_Uninstall.sh bin make_gss_instance sc_event.dtd
Logs global nvpair.dtd sc_reply.dtd
bash-3.00# ./Global_Gss_Uninstall.sh
Uninstallation Process will remove shared directories, installation
directories....
Do You Want To Proceed With Uninstallation [y/n]: [n] ? y
```

**Step 7** Enter **y (yes)** to continue the uninstall and start the cleanup operation.

```
Starting cleanup process
Cleaning the starentGss package ...
The following package is currently installed:
StarentGss Sun Cluster resource type for Gss server (sparc) 3.0.0,REV=
Do you want to remove this package? [y,n,?,q] y
```

**Step 8** Enter **y (yes)** to continue.

```
## Removing installed package instance <StarentGss>
This package contains scripts which will be executed with super-user
permission during the process of removing this package.
Do you want to continue with the removal of this package [y,n,?,q] y
```

**Step 9** Enter **y (yes)** to continue.

```
## Verifying package <StarentGss> dependencies in global zone
## Processing package information.
## Executing preremove script.
Resource <Gss-hars> has been removed already
Removing the resource type <Starent.Gss> ...
scrgadm -r -t Starent.Gss
done.
Network Resource not removed...
You may run removeGss again with the -h option to remove network
resource.
## Removing pathnames in class <none>
/opt/StarentGss/util/stopGss
```

```
/opt/StarentGss/util/startGss
/opt/StarentGss/util/removeGss
/opt/StarentGss/util/Gss_config
/opt/StarentGss/util
/opt/StarentGss/man/man1m/stopGss.1m
/opt/StarentGss/man/man1m/startGss.1m
/opt/StarentGss/man/man1m/removeGss.1m
/opt/StarentGss/man/man1m/Gss_config.1m
/opt/StarentGss/man/man1m
/opt/StarentGss/man
/opt/StarentGss/etc/Starent.Gss
/opt/StarentGss/etc
/opt/StarentGss/bin/simple_probe
/opt/StarentGss/bin/hasp_check
/opt/StarentGss/bin/gettime
/opt/StarentGss/bin/gethostnames
/opt/StarentGss/bin/Gss_validate.ksh
/opt/StarentGss/bin/Gss_update.ksh
/opt/StarentGss/bin/Gss_svc_stop.ksh
/opt/StarentGss/bin/Gss_svc_start.ksh
/opt/StarentGss/bin/Gss_probe.ksh
/opt/StarentGss/bin/Gss_mon_stop.ksh
/opt/StarentGss/bin/Gss_mon_start.ksh
/opt/StarentGss/bin/Gss_mon_check.ksh
/opt/StarentGss/bin
/opt/StarentGss/README.Gss
/opt/StarentGss
## Executing postremove script.
## Updating system information.
```

```
Removal of <StarentGss> was successful.
```

```
Global GSS uninstalled ...
```

If the following message is displayed when **Global\_Gss\_Uninstall.sh** is run, then some instance which needs to be uninstalled is still present and then, Global instance needs to be uninstalled. To identify which instance needs to be uninstalled, please see */etc/gss/instance\_configfile*.

```
There are instance[s] configured on this system, please uninstall the
instances using StandaloneGSSUninstall.sh script in instance directories
before uninstalling the global GSS installation
```

**Step 10** Login to node 2 and repeat *step 1* through *step 9* to uninstall the different instances of GSS and global instance from node 2.

## Upgrading multiple GSS Instances - Stand-alone Node

This process upgrades both the GSS and Postgres database application files for multiple GSS instances on a stand-alone node. The script begins by stopping all active processes and then restarts them after the upgrade is completed.

This upgrade process is valid for upgrading:

- GSS servers loaded with 7.1.x to GSS 8.0.x, or
- GSS servers loaded with 7.1.x to GSS 8.1.x.

### Preparing to Upgrade

**Step 1** Login to the GSS server.

**Step 2** Copy the current *gss.cfg*.

- Change to */<clus\_install\_dir>/gss/etc* directory and make a copy of the current *gss.cfg*.
- Rename the file and store it in a separate directory. It is recommended to store the copy in the */<packages>/<gss\_version>* directory holding the current GSS version.

**Step 3** Move a copy of the new GSS version to the directory where you have stored other folders containing earlier version. Unzip the compressed file for the new GSS version by entering the following command:

```
gunzip gss_<version>.tar.gz
```

*<version>* is the version number of the GSS software distributed in the zip file. For example, *gss\_8.0.xx.tar.gz*

**Step 4** Locate the tar file *GSS\_<version>.tar* in the */<packages>* directory and untar the file by entering the following command:

```
tar -xvf gss_<version>.tar
```

During the untar process, a */gss\_<version>* directory (for example: */gss\_8.0.xx*) is created in the the */<packages>* directory.

## Using the Installation Script


Installation is accomplished using the `inst_serv` script. It provides a menu-driven interface with question prompts. Most prompts display default values or information derived from the server's current setup - such as IP addresses for configured interfaces.

The following information will help you use the installation script most effectively:

- **Ctrl-C** will abort the installation process at any time during the procedure.
- The script senses whether the server is a stand-alone node or a cluster node.
- To accept defaults and continue the process, enter **n** as your choice for any prompt.
- To change a displayed value, enter the menu item number to display an entry field or a list of possible values.

The information from the prompts is used to generate the GSS configuration file (`gss.cfg`). This file can be changed at anytime after the installation.

---

 **Important:** It is recommended that you fill in path prompts only after you have created the directories to be used.

---

## Upgrading a GSS Stand-alone Node


The following procedure assumes that you are logged in to the GSS server with root privileges and that you are starting at the root directory level.

**Step 1** Change to the `/<packages>/gss_<version>` directory that was created when the files were uncompressed.

**Step 2** Locate the installation script file `inst_serv` and execute the following command:

```
./inst_serv
```

---

 **Important:** This script will check the version of the operating system installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Stand-alone Deployment* section, the script will abort the GSS installation.

---

The following appears, with pauses for validation, after entering the `inst_serv` command:

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
```

required changes in '/etc/system' file, restart the machine to get these changes reflected and then start installation again.

```
set msgsys:msginfo_msgmnb=65536
set msgsys:msginfo_msgtql=1024
set shmsys:shminfo_shmmax=33554432
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=256
set shmsys:shminfo_shmseg=256
set semsys:seminfo_semmmap=256
set semsys:seminfo_semmni=512
set semsys:seminfo_semmns=512
set semsys:seminfo_semmsl=32
```

Standalone Mode Installation [in standalone environment]...

GTPP Storage Server installation directory

Path where Gss will be installed :

GSS Installation dir ? /<install\_dir>



**Important:** If the upgrade is to a GSS 8.0.xx then /opt will appear as a default value for the **GSS Installation dir** parameter.

**Step 3** Enter the name of the directory where the new GSS is to be installed.

```
</install_dir>
```

**Step 4** After entering the name of the directory the script senses that GSS is already installed and prompts to upgrade.

```
Do you want to upgrade GSS [n] ? y
```

**Step 5** Enter y (yes) to continue the upgrade.

```
Starting Standalone upgrade
```

```
This will stop Process Monitor Tool along with GSS and Filegen
```

```
Please see log/psmon.log file for log messages
```

```
Stopping Process Monitor Tool...
```

```
Done.
```

```
Stopping GSS...
```

```

Done.

Stopping GSS FileGen...

Process GSS FileGen could not be stopped..

Killing it...

Done.

waiting for server to shut down.... done

server stopped

server starting

Starting database upgrade...

Database upgrade completed.....

waiting for server to shut down.... done

server stopped

Do You Want To Start GSS [y/n]: [y] ?

```

**Step 6** Enter **y** or press **Enter** to start the GSS and all related processes (e.g. the Process Monitor).

This will start Process Monitor Tool along with GSS and Filegen using params listed in `<install_dir>/gss/etc/gss.cfg`

Please see `log/psmon.log` file for log messages

Starting Process Monitor Tool...

Done.

Capturing status, please wait for a while...

```

=====
0 17453 11:35:15 TS 59 0:00 /<install_dir>/gss/lib/perl5.8.5
/bin/perl -w /<install_dir>/gss/psmon --daemon --cron 1
0 17460 11:35:16 TS 59 0:01 /<install_dir>/gss/bin/gss 1
0 17465 11:35:16 TS 59 0:01 /<install_dir>/gss/bin/gssfilegen 1
0 17108 11:33:08 TS 59 0:00 tee -a /<install_dir>/gss_7_1_67
/installation_log_<install_dir>_gss1 17067
0 2791 Oct_13 TS 59 1:01 /bin/bash /<install_dir>/gss/bin
/test_process.sh 1

```

```

0 17421 11:35:12 TS 59 0:00 tee -a /</install_dir>/gss_7_1_67/
installation_log_</install_dir>_solleims 17106
postgres 17479 17435 0 11:35:16 ? 0:00 /</install_dir>/postgres/
bin/postgres -D /</install_dir>/postgres/data -i
postgres 17485 17435 0 11:35:16 ? 0:00 /</install_dir>/postgres/
bin/postgres -D /</install_dir>/postgres/data -i
...
...
...
postgres 17491 17435 0 11:35:16 ? 0:00 /</install_dir>/postgres/
bin/postgres -D /</install_dir>/postgres/data -i
=====
GTPP Storage Server Version <gss_x_x_xx> installation done.
=====

```

The status display indicates that GSS, FileGen, PSMON, and PostgreSQL have all been started. If nothing displays, turn to the *Troubleshooting the GSS* section in the *GTPP Storage Server Administration* chapter.

## Upgrading multiple GSS instances - Cluster Nodes

This process upgrades both the GSS and Postgres database application files for multiple GSS instances on the nodes in a cluster.

This upgrade process is valid for upgrading:

- GSS servers loaded with 7.1.x to GSS 8.0.x, or
- GSS servers loaded with 7.1.x to GSS 8.1.x

### Preparing to Upgrade

**Step 1** Login to the GSS server.

**Step 2** Copy the current *gss.cfg*.

- Change to /<clus\_install\_dir>/gss/etc directory and make a copy of the current *gss.cfg*.
- Rename the file and store it in a separate directory. It is recommended to store the copy in the /<packages>/<gss\_version> directory holding the current GSS version.

- Step 3** Move a copy of the new GSS version to the directory where you have stored other folders containing earlier version. Unzip the compressed file for the new GSS version by entering the following command:

```
gunzip gss_<version>.tar.gz
```

<version> is the version number of the GSS software distributed in the zip file. For example, *gss\_8.0.xx.tar.gz*

- Step 4** Locate the tar file *GSS\_<version>.tar* in the */<packages>* directory and untar the file by entering the following command:

```
tar -xvf gss_<version>.tar
```

During the untar process, a */gss\_<version>* directory (for example: */gss\_8.0.xx*) is created in the the */<packages>* directory.

## Upgrading Node1 - Primary Node

By using the upgrade procedure, you ensure that no CDRs are lost as there will always be one of the nodes in active/online mode.

The primary node in a cluster is typically referred to as node1. You should begin the process by logging in to the node1.

- Step 1** Change to the directory where the untarred files were stored and initiate the installation script with the following command:

```
./inst_serv
```



**Important:** This script will check the version of operating system and cluster software installed on the system. If it is not matching the requirements in the *Minimum System Requirements for Cluster Deployment* section, the script will abort the GSS installation.

The following appears, with pauses for validation, after entering the **inst\_serv** command:

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
required changes in '/etc/system' file, restart the machine to get these
changes reflected and then start installation again.
```

```
set msgsys:msginfo_msgmnb=65536
```

```
set msgsys:msginfo_msgtql=1024
```

```

set shmsys:shminfo_shmmax=33554432
set shmsys:shminfo_shmmin=1
set shmsys:shminfo_shmmni=256
set shmsys:shminfo_shmseg=256
set semsys:seminfo_semmmap=256
set semsys:seminfo_semmni=512
set semsys:seminfo_semmns=512
set semsys:seminfo_semmsl=32

```

Cluster mode GSS installation exists /<server\_name>

**Step 2** The script registers that GSS is already installed and prompts you to upgrade the GSS installation. Enter **y** to upgrade the GSS application to a higher version.

```
Do you wish to upgrade this installation [y/n]: [n] ?
```

The following prompt appears only when the backup mode has not been configured during the fresh installation of GSS.

```
Do you want Backup installation for current cluster mode installation
[y/n]: [n] ?
```

```
*** PostgreSQL configuration for backup Installation ***
```

- 1) PostgreSQL port for backup installation :
- 2) PostgreSQL login for backup installation :
- 3) PostgreSQL data directory for backup installation :
- n) Proceed to next configuration
- a) Abort Installation

```
Enter Your Choice : [n] ?
```

**Step 3** After responding to continue the upgrade, the script displays upgrade requirements.

Please note that GSS upgrade in cluster mode needs following steps:-


1. Database upgrade on either of the nodes.
2. The node should be in standalone mode (after DB upgrade is done).

```
-----
```

This procedure will walk you through database upgrade first and then boot the node in standalone mode.

**Step 4** Indicate if you want to upgrade the database, press **Enter** or type **y**.

---

 **Important:** The database can only be upgraded on the node where the GSS is currently online (primary).

---

```
Do you want to upgrade database? (If you have already upgraded it on
another node, you may answer no to this question and proceed to boot the
node in standalone mode.) [y/n] : [y] ? y
```

The script proceeds to inform you of the next actions that are being taken.

```
Resource group "Gss-harg" is online on node "clustgss2".
```

```
For DB upgrade you need to start/switch resourcegroup on node clustgss1.
```

```
Do you want to start/switch resourcegroup "Gss-harg" on node clustgss1.
[y/n] : [y] ?
```

```
Bringing resource group online on node clustgss1, please wait...
```

```
Using "cluster_db_upgrade" script from "<packages>/gss_<version>" to
upgrade database.
```

```
clustgss
```

```
Starting database upgrade...
```

```
Database upgrade completed successfully.....
```

```
You need to reboot the node in standalone mode to upgrade the GSS
```

```
After rebooting node in standalone mode, use "inst_serv" script for GSS
upgrade.
```

```
Do you want to reboot the node in standalone mode [y/n] : [y] ?
```

**Step 5** Press **Enter** or type **y** to reboot into stand-alone mode and continue the upgrade process.

```
Rebooting node .....
```

**Step 6** Login to node2 as root and confirm the switch of the primary node (node1) to standby and the standby (node2) to active mode.

```
login as: root
```

```
# clresourcegroup
status
```

```
=== Cluster Resource Groups ===
```

Group Name	Node Name	Suspended	Status
Gss-harg	<name_node2>	No	Online
Gss-harg	<name_node1>	No	Offline

The display above indicates the upgrade script successfully performed the switchover.



**Important:** It will take a minute or two for the resource group to switch to node2 and for node1 to reboot in stand-alone mode.

**Step 7** Confirm node1 has rebooted in stand-alone mode by moving to node1 and entering the following command:

```
scstat
```

The system should indicate the node is not a cluster node.

```
scstate: not a cluster member.
```

**Step 8** Return to node1 and move to the directory where the GSS application files were stored, `/<packages>/gss_<version>` and initiate the installation script.

```
# ./inst_serv
```

```
Checking For Root Privileges .....
```

```
Done
```

```
Warning :
```

```
Before starting installation process, please make sure that intended
postgres username does not exist.
```

```
During "cluster mode" installation process, postgres user will be created
with UID 100001. Before starting cluster mode installation, please make
sure that UID 100001 is not in use.
```

```
Please check that the following parameters are set in the '/etc/system'
file. If they are not, please abort the installation using ^C , make
required changes in '/etc/system' file, restart the machine to get these
changes reflected and then start installation again.
```

```
set msgsys:msginfo_msgmnb=65536
```

```
set msgsys:msginfo_msgtql=1024
```

```
set shmsys:shminfo_shmmax=33554432
```

```
set shmsys:shminfo_shmmin=1
```

```
set shmsys:shminfo_shmmni=256
```

```
set shmsys:shminfo_shmseg=256
```

```
set semsys:seminfo_semmmap=256
```

```
set semsys:seminfo_semmni=512
```

```
set semsys:seminfo_semmns=512
```

```
set semsys:seminfo_semmsl=32
```

```
Cluster mode GSS installation exists /<clus_install_dir>
Do you wish to upgrade this installation [y/n]: [n] ?
```

**Step 9** Press **Enter** or type **y (yes)** to continue the upgrade.

```
Starting upgrade of cluster

Creating backup of previous installation files as
/<gss_install_dir>/gss_backup.tar

Backup done.

GSS Cluster Agent Installation configuration
Proceeding with GSS Cluster Agent Installation

Extracting StarentGss.tar...

Processing package instance <StarentGss> from </<package_dir>
/gss_<x_x_xx>

Sun Cluster resource type for Gss server (sparc) 3.0.0,REV=
Sun Microsystems, Inc.

This appears to be an attempt to install the same architecture and
version of a package which is already installed. This installation will
attempt to overwrite this package.

Using </opt> as the package base directory.

## Processing package information.
## Processing system information.

 28 package pathnames are already properly installed.

## Verifying package dependencies.
## Verifying disk space requirements.
## Checking for conflicts with packages already installed.
## Checking for setuid/setgid programs.

This package contains scripts which will be executed with super-user
permission during the process of installing this package.

Do you want to continue with the installation of <StarentGss> [y,n,?] y
```

**Step 10** Press **Enter** or type **y** to confirm the upgrade of the cluster resources.

```
Installing Sun Cluster resource type for Gss server as <StarentGss>
## Installing part 1 of 1.
```

```

[ verifying class <none> ]
## Executing postinstall script.
Installation of <StarentGss> was successful.
Extracting perl tar... Done.
Starting Backup database upgrade...
Backup Database upgrade completed.....
waiting for server to shut down.... done
server stopped

You need to reboot the node in cluster mode for normal cluster mode
operation.

If database is not upgraded on another node, please run
"cluster_db_upgrade" script on another node for database upgrade.

Do you want to reboot the node in cluster mode [y/n] : [y] ?

```

- Step 11** Upgrade on node1 is complete so you need to confirm node1 reboot so that node will return to cluster mode. Press **Enter** or type **y (yes)** to reboot.

```
Rebooting node .....
```

- Step 12** Confirm node1 has returned to the cluster. Generate a status display with the following command:

```

cnode status

=== Cluster Nodes ===

--- Node Status ---

Node Name          Status
-----
<node_name1>      Online
<node_name2>      Online

```

The system recognizes node1 as the standby cluster node.

- Step 13** Confirm the version of GSS installed on node1 with the following command:


```
./GSS -version
```



**Important:** `./serv version` is used when upgrading to GSS versions higher than x.x.69.

- Step 14** Switch the GSS resource group from node2 back to node1 with the following commands found in the `/<install_dir>/gss` directory.


---

 **Important:** Reminder from previous step - `/opt` is a default installation directory for some versions of GSS.

---

```
cd /opt/gss
./GSS switch
```

---

 **Important:** `./serv switch` is used when upgrading to GSS versions higher than x.x.69.

---

After entering the switchover command, the system displays the following:

```
Resource group "Gss-harg" is online on node "<name_node2>".
Bringing resource group online on node "<name_node1>". Please wait ...
Done.
```

**Step 15** Confirm the switchover was successful, node1 is again the primary node, with the following command and display:

```
clresourcegroup status


=== Cluster Resource Groups ===

Group Name          Node Name          Suspended   Status
-----
Gss-harg            <name_node1>      No          Online
                   <name_node2>      No          Offline
```

Now GSS resource group is active with the new release and node2 is in standby mode and free to be upgraded.

**Step 16** Perform *step 1 to step 11* on node2 to upgrade it to the newer version of GSS.

---

 **Important:** While executing *step 3* on the second node, do not choose to upgrade database, as it has already upgraded for first node. After completing the steps on node2, GSS upgrade on cluster setup will be completed.

---

## Configuring IPMP on GSS Server (Optional)

With IPMP, two or more network interface cards (bge0, bge1 etc.) are dedicated for each network to which the host connects. Each interface is assigned a static “test” IP address, which is used to access the operational state of the interface. Each virtual IP address is assigned to an interface, though there may be more interfaces than virtual IP addresses, some of the interfaces being purely for standby purposes. When the failure of an interface is detected its virtual IP addresses are swapped to an operational interface in the group.

The IPMP load spreading feature increases the machine's bandwidth by spreading the outbound load between all the cards in the same IPMP group.



**Important:** IPMP is a feature supported on Sun® Solaris® provided by Sun Microsystems. The configuration is included in the *System Administration Guide*. For more information, refer to the *Sun documentation*.

This section describes the following procedures to configure IP Multipathing on GSS server:

- [Configuring Probe-based IP Multipathing](#)
- [Configuring Link-based IP Multipathing](#)

Before proceeding for IPMP configuration here are some terms related to IPMP configuration:

- **Multipath Interface Group:** This the name given to the group of network devices in a multipath configuration.
- **Test Addresses:** These are IP addresses assigned to each board/interface of the multipath group, they do not move but should not be used for connections in or out of the host.
- **Multipath/float Address:** This is the IP address allocated to a Multipath Interface Group that is shared between all devices in the group (either by load sharing or active standby).

### Configuring Probe-based IP Multipathing

Configuration procedure given here assumes that:

- GSS Server Host name is `<hostname>`
- `<NIC_1>` and `<NIC_2>` are the network interface devices; i.e. bge0, bge1 etc.
- Using network device `<NIC_2>` as active and `<NIC_1>` as the Standby.
- Multipath Interface Group name is `<multipath_grp>`
- Multipath IP address is `<multipath_IP_address>`
- Test IP address for `<NIC_1>` interface in `<test_IP_address_NIC_1>`
- Test IP address for `<NIC_2>` interface in `<test_IP_address_NIC_2>`

**Step 1** Ensure that the MAC addresses on the host are unique by setting the local-mac-address parameter to true by running following command as `root` user:

```
eeeprom local-mac-address?=true
```

**Step 2** Create an `<NIC_1>` for the Standby network device with the following entry:

```
<hostname>-<NIC_1> netmask <netmask> broadcast+group <multipath_grp> deprecated -
failover standby up
```

<hostname> is name of the Host and <NIC\_1> is the network device to be set as Standby.

<multipath\_grp> is Multipath Interface Group name given to the group of network devices in a multipath configuration.

<netmask> is the sub-netmask used by network.

**Step 3** Create an `/etc/hostname.<NIC_2>` for the active network device with the following entry:

```
<hostname>-<NIC_2> netmask 255.255.255.0 broadcast+group <multipath_grp>deprecated
-failover up addif <hostname>-active netmask 255.255.255.0 broadcast+failover up
```

<hostname> is name of the Host and <NIC\_2> is the network device to be set as active.

<multipath\_grp> is Multipath Interface Group name given to the group of network devices in a multipath configuration.

**Step 4** Edit the `/etc/hosts` file using “vi editor” and add the following three entries:

```
<multipath_IP_address> <hostname>-active
<test_IP_address_NIC_1> <hostname>-<NIC_1>
<test_IP_address_NIC_2> <hostname>-<NIC_2>
```

`multipath_IP_address` is the IP address allocated to a Multipath Interface Group that is shared between all devices in the group (either by load sharing or Active-Standby).

`test_IP_address_NIC_1` is the IP addresses assigned to <NIC\_1> interface of the multipath group, they do not move but should not be used for connections in or out of the host.

`test_IP_address_NIC_2` is the IP addresses assigned to <NIC\_2> interface of the multipath group, they do not move but should not be used for connections in or out of the host.

**Step 5** Restart the host by entering the following command:

```
shutdown -i 6 -g 0 -y
```

## Configuring Link-based IP Multipathing

Configuration procedure provided here assumes that:

- GSS Server Host name is <hostname>
- <NIC\_1> and <NIC\_2> are the network interface devices; i.e. bge0, bge1 etc.
- Using network device <NIC\_1> as active and <NIC\_2> as the standby
- Multipath Interface Group name is <multipath\_grp>
- Multipath IP address is <multipath\_IP\_address>
- Test IP address for <NIC\_1> interface in <test\_IP\_address\_NIC\_1>
- Test IP address for <NIC\_2> interface in <test\_IP\_address\_NIC\_2>
- <my\_address> is associated with Multipath IP address <multipath\_IP\_address> in the `/etc/hosts` file

**Step 1** Ensure that the MAC addresses on the host are unique by setting the **local-mac-address** parameter to true by running following command as *root* user:

```
eeeprom local-mac-address?=true
```

**Step 2** Create an `/etc/hostname.<NIC_1>` for the Active network device with the following entry:

```
<my_address> netmask + broadcast + group <multipath_grp> up
```

*<my\_address>* is associated with Multipath IP address *<multipath\_IP\_address>* in the `/etc/host` file.

*<multipath\_grp>* is Multipath Interface Group name given to the group of network devices in a multipath configuration.

**Step 3** Create an `/etc/hostname.<NIC_2>` for the Standby network device with the following entry:

```
group <multipath_grp> up
```

**Step 4** Restart the host by entering the following command:

```
shutdown -i 6 -g 0 -y
```




# Chapter 3

## GTPP Storage Server Administration

---

This chapter provides information for the administration of the GSS; specifically maintenance and operation procedures and troubleshooting procedures. In most cases, the maintenance and troubleshooting procedures for a cluster node are the same or similar to those for a stand-alone node - when there are differences there will be separate procedures for stand-alone and cluster.

---

 **Important:** The GSS is not configured or managed through network management interface(s).

---

This chapter includes procedures for maintaining and troubleshooting the GSS software application.

- [Maintaining the GSS](#)
- [Troubleshooting the GSS](#)

## Maintaining the GSS

This section provides the procedures on how to maintain the GSS application and how to use the GSS configuration and log files.

### Periodic Removal of Processed Data Files

The GSS can be configured to remove the processed CDR data files. The removal can be done periodically and automatically through a clean-up script (`<install_dir>/gss/bin/cleanup.sh`) which can be run as a cron job.

The cron job is a crontab command. It is used to schedule commands to be executed periodically. It reads a series of commands from standard input and collects them into a file known as crontab file which is later read and whose instructions are carried out.

The following procedure adds a clean-up entry to the crontab file.

**Step 1** Login to the server with root privileges.

**Step 2** Enter the editing mode for the crontab with the following command:

```
crontab -e
```

**Step 3** Add the job execution frequency values as required.

```
00 03 * * * * /<install_dir>/gss/bin/cleanup.sh -ext .p -dirpath
<install_dir>/gss/data/extradata -clnintvl 7 >> /gss/log/cleanup.log 2>&1
```

Adding the above entry to the crontab table instructs the system to execute the clean-up script every day at 3:00 AM and to delete all the files with a `.p` file extension that are older than the maximum storage period, 7 days.

```
00 03 * * * * /<install_dir>/gss/bin/cleanup.sh -ext .p -dirpath
/sharedgss/gss/clustgss1_log/gss/ -clnintvl 7 >>
/sharedgss/gss/clustgss1_log/cleanup.log 2>&1
```

Add the above entry to a crontab instructs the system to execute the clean-up script to remove processed data files in the `/sharedgss/gss/data` directory after the expiration of the storage period of 7 day(s).

**Step 4** Save and close the crontab file by entering the following command:

```
:wq
```



**Important:** The command suggested for this step assumes that you have the `vi` editor set as the system's default text editor, if not then use the appropriate exit command.

**Step 5** View the changes in the `crontab` file by entering the following command:

```
crontab -l
```

## Using Log Files for Monitoring

Each component and most processes generate their own log files to track activity, errors and warning messages. Use the `less` command or a text editor to view the appropriate log.




 **Important:** Information inside angle brackets denotes a variable that you defined during the installation process.

Table 2. GSS System Logs

Application or Process	Log File Location
Installation	Location: /<packages>/gss_<version>/ Log name format: installation_log_<install_dir>_<server_name> Log name example: installation_log_install8071_LondonGSS1
GSS server - Stand-alone mode	Location: / <install_dir>/gss/log/GSS_SERVER_LOG_<YYYYMMDD>_<HHMMSS>/GSS_SERVER_LOG Log directory name example: GSS_SERVER_LOG_20081124_020635 Log name: GSS_SERVER_LOG
GSS server - Cluster mode	Location: /sharedgss/gss/<node_name>_log/GSS_SERVER_LOG_<YYYYMMDD>_<HHMMSS>/GSS_SERVER_LOG Log directory name example: GSS_SERVER_LOG_20081124_030851 Log name: GSS_SERVER_LOG
PostgreSQL	<install_dir>/postgres/logfile
PSMON	<install_dir>/gss/log/psmon.log
FileGen - Stand-alone mode	<p>FileGen logs are stored in the same directory as the GSS application logs. They are distinguishable by the log name.</p> <hr/> <p> <b>Important:</b> The FileGen adds an audit message in syslog file located in <code>/var/adm/messages</code> directory for every CDR file being generated. Log level <code>user.notice</code> should be turned on in <code>/etc/syslog.conf</code> file for syslogd to log these messages.</p> <hr/> <p>Location: / &lt;install_dir&gt;/gss/log/GSS_SERVER_LOG_&lt;YYYYMMDD&gt;_&lt;HHMMSS&gt;/GSS_FILEGEN_LOG Log directory name example: GSS_SERVER_LOG_20081124_020635 Log name: GSS_FILEGEN_LOG</p>
FileGen - Cluster mode	<p>FileGen logs are stored in the shared location.</p> <hr/> <p> <b>Important:</b> The FileGen adds an audit message in syslog file located in <code>/var/adm/messages</code> directory for every CDR file being generated. Log level <code>user.notice</code> should be turned on in <code>/etc/syslog.conf</code> file for syslogd to log these messages.</p> <hr/> <p>Location: /sharedgss/gss/&lt;node_name&gt;_log/GSS_SERVER_LOG_&lt;YYYYMMDD&gt;_&lt;HHMMSS&gt;/GSS_FILEGEN_LOG Log directory name example: GSS_SERVER_LOG_20090525_130248 Log name: GSS_FILEGEN_LOG</p>

## Accessing GSS Commands

There are several console commands available at the command line. Access is gained by moving to the `/gss` directory where the GSS files were installed.

```
# cd /<install_dir>/gss
```


Depending upon the version of the GSS software, the root command will vary. For all versions with an extension number of 70 or lower (e.g., 7.1.67 or 8.0.70) the command is



```
# ./GSS
```

For all versions of GSS with an extension number of 71 or higher, the command is

```
# ./serv
```

**Table 3. Command Arguments**

Action	Meaning
start	Starts the GSS, Process Monitor (PSMON) Tool, PostgreSQL server and FileGen using parameters listed in <code>gss.cfg</code> file.
stop	Stops the GSS, PSMON Tool, PostgreSQL server and FileGen utility.
restart	Stops and restarts the GSS, PSMON Tool, PostgreSQL server and FileGen utility. While restarting the GSS, a confirmation message appears both in the case of stand-alone and cluster modes.
switch	<p>Switchover resource group from one node to another.</p> <hr/> <p> <b>Important:</b> The <code>switch</code> option can only be used in a cluster mode installation.</p>
status	<p>Displays the current status of each process/resource.</p> <p>In case of stand-alone/backup mode, this command displays the status of the following processes:</p> <ul style="list-style-type: none"> <li>• Process monitor application</li> <li>• PostgresGSS server</li> <li>• GTPP Storage Server</li> <li>• GSS filegen</li> </ul> <p>In case of cluster mode, this command displays the status of the following resources:</p> <ul style="list-style-type: none"> <li>• GSS resource group</li> <li>• Failover Dataservice resource</li> <li>• Logical hostname resource</li> <li>• CRNP resource</li> </ul>

Action	Meaning
start_backup	Starts the GSS in backup mode.   <b>Important:</b> The <code>start_backup</code> option can only be used in a backup mode installation.
stop_backup	Stops the GSS in backup mode.   <b>Important:</b> The <code>stop_backup</code> option can only be used in a backup mode installation.
--version	Displays the version of the GSS software. Only used with <code>GSS</code> command.
version	Displays the version of the GSS software. Only used with <code>serv</code> command.
help	Prints this help message.

The commands are used in the following manner:

```
# ./serv switch
# ./serv start
# ./GSS --version
```

In the stand-alone mode, the status of individual process can be identified with `serv` script by entering the following command:

```
# ./serv status <psmon | postgres | gss | gssfilegen>
or
# ./serv <psmon | postgres | gss | gssfilegen> status
```


In the cluster mode, the status of individual resource and resource group can be identified with `serv` script by entering the following command:

```
# ./serv status [ <resource_name> | <resourcegroup_name> | <process_name> ]
or
# ./serv [ <resource_name> | <resourcegroup_name> | <process_name> ] status
```

For example, if you want to start only the PSMON using `serv` script, then enter the following command:

```
# ./serv psmon status
or
# ./serv status psmon
```

---

 **Important:** For 9.0.106 and later releases, the `./serv status` command displays Partial Filegen and Final Filegen status only for GTPP Dictionary custom21.

---

## Modifying a GSS Configuration

The GSS configuration file (**gss.cfg**) is stored in the `/<install_dir>/gss/etc` directory.

The **gss.cfg** file can be modified with any text editor but for the modifications to take effect, the server must be rebooted.



**Caution:** Before making changes to a **gss.cfg**, ensure that there is a write-protected copy of the **gss.cfg** stored in a separate directory on each node. It is recommended to store the copy in the `/<packages>/<gss_version>` directory.

---

The following are the copies of **gss.cfg**. Besides providing the default GSS configuration values, this information can be copied to replace a damaged **gss.cfg**.

To view a **gss.cfg**, use the following command:

```
cat /<install_dir>/gss/etc/gss.cfg
```

### GSS Configuration File - Stand-alone Node

```
#Filename: gss.cfg

#Config file containing configuration values for the GTPP Storage Server (GSS)
and Filegen process organised with a Section header and Key = Value data format.

#Section header [Tag string in sqr bracket] is for internal use.

#User can update the Value part and restart the process making it effective.

#All numerical values have Range and Default.

#Section: Server_Param

#Contains config values for Server like address, port, exec-mode etc.

[Server_Param]

#Key: Addr

#A valid existing host-interface address.

#Default: 0.0.0.0

Addr = 10.4.4.132

#Key: Port

#A valid udp-port value for Server process.

#Filgen process uses Port + 1 value.
```

```
#Range : 1024 - 65535
#Default: 50000
Port = 50000

#Key: DB_Init_Tables
#Clean up all tables while starting the server.
#Caution: This will delete all the records from database!!!
#Value : yes / no
#Default: no
DB_Init_Tables = no

#Key: Socket_RX_Buffer_Size
#Socket Rx Buffer Size in Bytes.
#Consider increasing Socket buffer size at system kernel level and then at
#application level for GSS if netstat -s (udpInOverflows) shows non zero
incremental values for large messages.
#To increase overall kernel limit use (as root):
#ndd -set /dev/udp udp_max_buf 8388608/16777216 or higher:limited by system RAM
#And then set the same value with Socket_RX_Buffer_Size for Server.
#Note :This is applicable only for GSS Process. Filegen uses default value.
#Range : 32768 - 268435456
#Default: 16777216 Bytes
Socket_RX_Buffer_Size = 16777216

#Key: DB_Connections
#GSS process Database Connections count.
#Range : 12 - 32
#Default: 20
DB_Connections = 20
```

```
#Key:Archive_Mode

#Run Server and Filegen in archive mode where each data request is archived
directly after Store w/o Update (Ack) from CGF.

#To be used if GSN is configured without any CGF but uses Storage Server as the
only backup storage node.

#Value : yes/no

#Default: yes

#Note: By default Archive_Mode is set to yes (GSS only), please do not change
this value as Normal Mode (Archive_Mode = no) is Obsoleted. GSS + CGF
configuration is no longer in use and is Obsoleted.

Archive_Mode = yes

#Section: Filegen_Param

#Contains config values for Filegen parameters.

[Filegen_Param]

#Key: Start_Seq_Num

#Starting seq num of the Ist CDR file.

#Uses this number for the first file or

#if Use_Start_Seq_Num is set.

#Range : 1 - ULONG_MAX

#Default: 1

Start_Seq_Num = 1

#Key: Use_Start_Seq_Num

#Flag to indicate whether to use the Start_Seq_Num parameter for the Ist CDR
file.

#If yes, the Start_Seq_Num value is used for the Ist CDR file on restart.

#Value : yes/no

#Default: no

Use_Start_Seq_Num = no
```

```
#Key: Start_Lrsn_Num
#Starting lrsn num of the Ist CDR file.
#Uses this number for the first file or
#if Use_Start_Lrsn_Num is set.
#Range : 1 - ULONG_MAX
#Default: 1
Start_Lrsn_Num = 1

#Key: Use_Start_Lrsn_Num
#Flag to indicate whether to use the Start_Lrsn_Num parameter for the Ist CDR
file.
#If yes, the Start_Lrsn_Num value is used for the Ist CDR file on restart.
#Value : yes/no
#Default: no
Use_Start_Lrsn_Num = no

#Key: Max_CDR_Per_File
#Max count of CDR in a data file
#Range : 1000 - 100000
#Default: 10000
#Note : Generated data file may contain 1 to 254 CDRs more in addition to
Max_CDR_Per_File
Max_CDR_Per_File = 10000

#Key: Max_File_Gen_Period
#Maximum idle period in hours after which a new file shall be generated without
waiting for Maximum CDRs per file
#This timer is reset on receiving CDRs or writing a new file.
#Range : 1 - 24 hours
#Default: 1 hour
```

```
Max_File_Gen_Period = 1

#Key: Hard_File_Gen_Period

#Hard limit interval in mins after which a new file shall be generated without
waiting for Maximum CDRs per file.

#Setting this hard limit also enables writing a new file of 0 CDRs on timer
expiry.

#This timer is reset on writing a new file.

#Overrides Max_File_Gen_Period if configured.

#Range : 1 - 24*60 mins

#Default: 0 (disable)

#Description: Hard Limit Interval for File Generation

Hard_File_Gen_Period = 0

#Key: Max_Req_Per_Fetch

#Maximum number of data records to be fetched per db request.

#Range : 10 - 1000

#Default: 50

Max_Req_Per_Fetch = 50

#Key: File_Format

#Encoding format for CDR files

#Values : starent/custom1/custom2/custom3/custom4/custom5/custom6/custom7

#Note : File format custom6 is reserved for HDD GSS on GSN side and not
supported for external GSS

#Default : starent

File_Format = custom3

#Key: GGSN_Location

#File name format string for identifying CDR files for a node/location.
```

```
#Note : This is applicable only for custom2, custom3 and custom4 file format.

#Even though name of this parameter contains GGSN, this parameter is equally
applicable to all type of GSNs supported by GSS.

#Values : GSN Location String

#Default : GSN

GGSN_Location = GSN

#Key: Data_File_Path

#Path to generated CDR data files.

#This has to be valid path with write permission to the directory.

#Note : For custom7 file format 3 types of files #(Processed,Filtered,Rejected)
gets generated in <gss_dir>/data, #<gss_dir>/data/filtered and
<gss_dir>/data/rejected respectively and #which are mapped to /cdr/work/proc,
/cdr/work/proc/filtered, #/cdr/work/proc/rejected respectively through symbolic
link

#Value : Dir path string. For example <gss_dir>/data

#Default: No default

Data_File_Path = /TEST_GSS/cvserver/gss/data

#Key: Enable_Redundant_File

#Flag to indicate whether to enable redundant file creation in path parallel to
primary data path. For example <gss_dir>/data_redundant

#Note : This is applicable only for
starent/custom1/custom2/custom3/custom4/custom5 file format

#This is not applicable for custom6 and custom7 file format

#Value : yes/no

#Default: no

Enable_Redundant_File = n

#Key: GTPP_Dict

#GTPP Dictionary to be used to decode and encode CDRs.

#The GTPP Dictionary should be same as used on GSN.
```

#Note : This is applicable only for custom2, custom3 and custom4 file format.

#Value : starent,custom1-custom20.

#Default: custom1

GTPP\_Dict = custom1

#Key: Use\_Binary\_Addr\_Format

#Flag to encode IP Address fields in binary format for the CDR.

#Note : This is applicable only for custom2, custom3 and custom4 file format.

#Value : yes/no

#Default: no

Use\_Binary\_Addr\_Format = n

#Key: Enable\_Unique\_LRSN

#Flag to encode unique incremental LRSN in the CDR.

#Note : This is applicable only for custom2, custom3 and custom4 file format.

#Value : yes/no

#Default: no

Enable\_Unique\_LRSN = n

#Key: Node\_ID

#Node ID string to be encoded in CDR.

#Note : This is applicable only for custom2, custom3 and custom4 file format.

#Value : No default value. To be encoded if user gives a Node ID string.

#Range : 0 - 16 characters

#Default : No default

Node\_ID =

#Key:Backup\_Sequence\_File\_Path

#Path for filegen file sequence used in backup mode

```
#Note : This is applicable only for
starent/custom1/custom2/custom3/custom4/custom5 file format

#This is not applicable for custom6 and custom7 file format

#Value : dir path string

#Default:

Backup_Sequence_File_Path =

#Section: Archive_Param

#Contains config values for archive parameters.

[Archive_Param]

#Key: Archive_Cdr

#Used for normal execution mode (GSN with CGF+GSS) to enable/disable archiving
CGF acked data requests (CDRs) to database.

#yes : GSS shall archive all Acknowledged requests to Archive DB.

#no : GSS shall delete all Acknowledged requests from Active DB without saving
them to Archive DB.

#Value : yes /no

#Default: yes

Archive_Cdr = yes

#Key: Req_Buffer_Length

#In memory internal data request buffer length maintained by server process.

#The count indicates number of data request in multiples of 512 data request
blocks.

#Range : 1 - 200

#Default: 2 (2*512=1024 data requests)

Req_Buffer_Length = 2

#Key: Max_Archive_Period

#Used for normal execution mode (GSN with CGF+GSS)
```

```

#Max archive period (hours) for which archived data
#requests (CDRs) are maintained with system's Archive DB.
#Range : 1 - 168 hours
#Default: 168 hours
Max_Archive_Period = 168

#Key: Max_Storage_Period
#Used for normal execution mode (GSN with CGF+GSS)
#Maximum storage period (mins) for which unacked data requests are maintained
with system after which are #transferred to Unacked Archive DB.
#A value of 0 indicates no such limitation and can be maintained forever.
#Range : 1 - 30 mins
#Default: 0 (Disable)
Max_Storage_Period = 0

#Key: Max_Storage_Count
#Used for normal execution mode (GSN with CGF+GSS)
#Maximum number of latest unacked data requests maintained by system in memory
internal data request buffer after which are transferred to Unacked Archive DB.
#Size in multiple of Req_Buffer_Length
#A value of 0 indicates no max value.
#Range : 1 - 1000
#Default: 10
Max_Storage_Count = 10

#Key: Vacuumize_DB
#Vacuumize database periodically.
#Value : yes / no
#Default: yes (Vacuumize database periodically)
Vacuumize_DB = yes

```

```
#Section: DB_Param
#Contains config values for db parameters.
[DB_Param]
#Key: DB_Host_Name
#DB server name or address string
#Default : No default
DB_Host_Name = 127.0.0.1

#Key: DB_User_Name
#User name for connection with database
#Value : User name string
#Default: postgres
#Description : PostgreSQL login
DB_User_Name = gsspg

#Key: DB_Password
#Password for connection with database
#Value : User name string
#Default: postgres
#Description :PostgreSQL passwd
DB_Password = gsspg

#Key: DB_Host_Addr
#DB server address.
#Default : No Default, if given overrides DB_Host_Name
DB_Host_Addr =

#Key: DB_Port
```

```
#DB server udp port value
#Range : 1 - 65535
#Default: 5432
#Description :PostgreSQL port
DB_Port = 5499

#Key: DB_Retry_Count
#Retry count to perform reconnection with DB server
#Range : 10 - 100
#Default: 20
DB_Retry_Count = 20

#Key: DB_Retry_Interval
#Retry Interval (secs) to perform reconnection with DB server
#Range : 1 - 10 secs
#Default: 2 secs
DB_Retry_Interval = 2

#Section: Notif_Param
#Contains config values for resource notification parameters.
[Notif_Param]
#Key: Notif_Type
#Value : Alarm / Alert

#Alarm : Generates notification once for each threshold cross over for the
monitored entity.

#Alert : Generates notification on each polling interval for the monitored
entity only if above threshold mark.

#Default: Alarm
Notif_Type = Alarm
```

```
#Key: Notif_Poll_Interval

#Notification poll interval (mins) to examine the state of the monitored entity.

#Range : 1 - 1440 mins

#If threshold configuration is disabled then default: 0 (Disable: No
notification)

#If threshold configuration is enabled then default: 1

Notif_Poll_Interval = 1

#Key: Notif_Retry_Interval

#Notification retry interval (secs)

#Range : 1 - 60 secs

#Default: 20 secs

Notif_Retry_Interval = 20

#Key: Notif_Retry_Count

#Notification retry count

#Range : 1 - 5

#Default: 3

Notif_Retry_Count = 3

#Key: Notif_GCDR_File_Period

#Enables Alarm / Alert to be sent for files older than given period in mins

#Note : Even though name of this parameter contains GCDR, this parameter is
equally applicable to all type of CDRs supported by GSS.

#Range : 1 - 7*24*60 mins

#Default: 15

Notif_GCDR_File_Period = 15

#Key: Notif_CPU_Usage

#Enables Alarm / Alert to be sent for system cpu usage in %.
```

```
#Range : 0 - 100 %
#Default: 60
Notif_CPU_Usage = 60

#Key: Notif_Mem_Usage
#Enables Alarm / Alert to be sent for system swap usage in MB.
#Range : 0 - ULONG_MAX (MB)
#Default: 9500
Notif_Mem_Usage = 9500

#Key: Notif_Disk_Usage
#Enables Alarm / Alert to be sent if available disk size(GB) falls below
configured limit.
#Range : 0 - ULONG_MAX (GB)
#Default: 199
Notif_Disk_Usage = 199

#Section: Log_Param
#Contains config values for logging and status.
[Log_Param]
#Key Show_Status
#Show status periodically (secs)
#Range : 0 - ULONG_MAX secs
#Default: 300 secs
#0 - Disables
Show_Status = 300

#Key: Print_IO_Msg
#Prints I/O messages to log
```

```
#Value : yes /no
#Default: no
Print_IO_Msg = no

#Key: Audit_File_Size
#Audit log file size
#minimum 2500 kbytes
#default 2500 kbytes
Audit_File_Size = 2500
```

## GSS Configuration File - Cluster Nodes

These files are the same for each node in a cluster.

```
#Filename: gss.cfg

#Config file containing configuration values for the GTPP Storage Server (GSS)
and Filegen process organised with a Section header and Key = Value data format.

#Section header [Tag string in sqr bracket] is for internal use.

#User can update the Value part and restart the process making it effective.

#All numerical values have Range and Default.

#Section: Server_Param

#Contains config values for Server like address, port, exec-mode etc.

[Server_Param]

#Key: Addr

#A valid existing host-interface address.

#Default: 0.0.0.0

Addr = 10.4.4.129

#Key: Port

#A valid udp-port value for Server process.

#Filgen process uses Port + 1 value.
```

```
#Range : 1024 - 65535
#Default: 50000
Port = 50000

#Key: DB_Init_Tables
#Clean up all tables while starting the server.
#Caution: This will delete all the records from database!!!
#Value : yes / no
#Default: no
DB_Init_Tables = no

#Key: Socket_RX_Buffer_Size
#Socket Rx Buffer Size in Bytes.
#Consider increasing Socket buffer size at system kernel level and then at
application level for GSS if netstat -s (udpInOverflows) shows non zero
incremental values for large messages.
#To increase overall kernel limit use (as root):
#ndd -set /dev/udp udp_max_buf 8388608/16777216 or higher:limited by system RAM
#And then set the same value with Socket_RX_Buffer_Size for Server.
#Note :This is applicable only for GSS Process. Filegen uses default value.
#Range : 32768 - 268435456
#Default: 16777216 Bytes
Socket_RX_Buffer_Size = 16777216

#Key: DB_Connections
#GSS process Database Connections count.
#Range : 12 - 32
#Default: 20
DB_Connections = 20
```

```
#Key:Archive_Mode

#Run Server and Filegen in archive mode where each data request is archived
directly after Store w/o Update (Ack) from CGF.

#To be used if GSN is configured without any CGF but uses Storage Server as the
only backup storage node.

#Value : yes/no

#Default: yes

#Note: By default Archive_Mode is set to yes (GSS only), please do not change
this value as Normal Mode (Archive_Mode = no) is Obsoleted. GSS + CGF
configuration is no longer in use and is Obsoleted.

Archive_Mode = yes

#Section: Filegen_Param

#Contains config values for Filegen parameters.

[Filegen_Param]

#Key: Start_Seq_Num

#Starting seq num of the Ist CDR file.

#Uses this number for the first file or

#if Use_Start_Seq_Num is set.

#Range : 1 - ULONG_MAX

#Default: 1

Start_Seq_Num = 1

#Key: Use_Start_Seq_Num

#Flag to indicate whether to use the Start_Seq_Num parameter for the Ist CDR
file.

#If yes, the Start_Seq_Num value is used for the Ist CDR file on restart.

#Value : yes/no

#Default: no

Use_Start_Seq_Num = no
```

```
#Key: Start_Lrsn_Num
#Starting lrsn num of the 1st CDR file.
#Uses this number for the first file or
#if Use_Start_Lrsn_Num is set.
#Range : 1 - ULONG_MAX
#Default: 1
Start_Lrsn_Num = 1

#Key: Use_Start_Lrsn_Num
#Flag to indicate whether to use the Start_Lrsn_Num parameter for the 1st CDR
file.
#If yes, the Start_Lrsn_Num value is used for the 1st CDR file on restart.
#Value : yes/no
#Default: no
Use_Start_Lrsn_Num = no

#Key: Max_CDR_Per_File
#Max count of CDR in a data file
#Range : 1000 - 100000
#Default: 10000
#Note : Generated data file may contain 1 to 254 CDRs more in addition to
Max_CDR_Per_File
Max_CDR_Per_File = 10000

#Key: Max_File_Gen_Period
#Maximum idle period in hours after which a new file shall be generated without
waiting for Maximum CDRs per file
#This timer is reset on receiving CDRs or writing a new file.
#Range : 1 - 24 hours
#Default: 1 hour
```

```
Max_File_Gen_Period = 1

#Key: Hard_File_Gen_Period

#Hard limit interval in mins after which a new file shall be generated without
waiting for Maximum CDRs per file.

#Setting this hard limit also enables writing a new file of 0 CDRs on timer
expiry.

#This timer is reset on writing a new file.

#Overrides
Max_File_Gen_Period if configured.

#Range : 1 - 24*60 mins

#Default: 0 (disable)

#Description: Hard Limit Interval for File Generation

Hard_File_Gen_Period = 2

#Key: Max_Req_Per_Fetch

#Maximum number of data records to be fetched per db request.

#Range : 10 - 1000

#Default: 50

Max_Req_Per_Fetch = 50

#Key: File_Format

#Encoding format for CDR files

#Values : starent/custom1/custom2/custom3/custom4/custom5/custom6/custom7

#Note : File format custom6 is reserved for HDD GSS on GSN side and not
supported for external GSS

#Default : starent

File_Format = custom3

#Key: GGSN_Location
```

```

#File name format string for identifying CDR files for a node/location.

#Note : This is applicable only for custom2, custom3 and custom4 file format.

#Even though name of this parameter contains GGSN, this parameter is equally
applicable to all type of GSNs supported by GSS.

#Values : GSN Location String

#Default : GSN

GGSN_Location = GSN

#Key: Data_File_Path

#Path to generated CDR data files.

#This has to be valid path with write permission to the directory.

#Note : For custom7 file format 3 types of files #(Processed,Filtered,Rejected)
gets generated in <gss_dir>/data, #<gss_dir>/data/filtered and
<gss_dir>/data/rejected respectively and #which are mapped to /cdr/work/proc,
/cdr/work/proc/filtered, #/cdr/work/proc/rejected respectively through symbolic
link

#Value : Dir path string. For example <gss_dir>/data

#Default: No default

Data_File_Path = /sharedgss/gss/data

#Key: Enable_Redundant_File

#Flag to indicate whether to enable redundant file creation in path parallel to
primary data path. For example <gss_dir>/data_redundant

#Note : This is applicable only for
starent/custom1/custom2/custom3/custom4/custom5 file format

#This is not applicable for custom6 and custom7 file format

#Value : yes/no

#Default: no

Enable_Redundant_File = n

#Key: GTPP_Dict

#GTPP Dictionary to be used to decode and encode CDRs.

```

```
#The GTPP Dictionary should be same as used on GSN.
#Note : This is applicable only for custom2, custom3 and custom4 file format.
#Value : starent,custom1-custom20.
#Default: custom1
GTPP_Dict = custom1

#Key: Use_Binary_Addr_Format
#Flag to encode IP Address fields in binary format for the CDR.
#Note : This is applicable only for custom2, custom3 and custom4 file format.
#Value : yes/no
#Default: no
Use_Binary_Addr_Format = n

#Key: Enable_Unique_LRSN
#Flag to encode unique incremental LRSN in the CDR.
#Note : This is applicable only for custom2, custom3 and custom4 file format.
#Value : yes/no
#Default: no
Enable_Unique_LRSN = n

#Key: Node_ID
#Node ID string to be encoded in CDR.
#Note : This
is applicable only for custom2 and custom3 file format.
#Value : No default value. To be encoded if user gives a Node ID string.
#Range : 0 - 16 characters
#Default : No default
Node_ID =
```

```
#Key:Backup_Sequence_File_Path

#Path for filegen file sequence used in backup mode

#Note : This is applicable only for
starent/custom1/custom2/custom3/custom4/custom5 file format

#This is not applicable for custom6 and custom7 file format

#Value : dir path string

#Default:

Backup_Sequence_File_Path =

#Section: Archive_Param

#Contains config values for archive parameters.

[Archive_Param]

#Key: Archive_Cdr

#Used for normal execution mode (GSN with CGF+GSS)

#to enable/disable archiving CGF acked data requests (CDRs) to database.

#yes : GSS shall archive all Acknowledged requests to Archive DB.

#no : GSS shall delete all Acknowledged requests from Active DB without saving
them to Archive DB.

#Value : yes /no

#Default: yes

Archive_Cdr = yes

#Key: Req_Buffer_Length

#In memory internal data request buffer length maintained by server process.

#The count indicates number of data request in multiples of 512 data request
blocks.

#Range : 1 - 200

#Default: 2 (2*512=1024 data requests)

Req_Buffer_Length = 2
```

```
#Key: Max_Archive_Period
#Used for normal execution mode (GSN with CGF+GSS)
#Max archive period (hours) for which archived data
#requests (CDRs) are maintained with system's Archive DB.
#Range : 1 - 168 hours
#Default: 168 hours
Max_Archive_Period = 168
```

```
#Key: Max_Storage_Period
#Used for normal execution mode (GSN with CGF+GSS)
#Maximum storage period (mins) for which unacked data requests are maintained
with system after which are transferred to Unacked Archive DB.
#A value of 0 indicates no such limitation and can be maintained forever.
#Range : 1 - 30 mins
#Default: 0 (Disable)
Max_Storage_Period = 0
```

```
#Key: Max_Storage_Count
#Used for normal execution mode (GSN with CGF+GSS)
#Maximum number of latest unacked data requests maintained by system in memory
internal data request buffer after which are transferred to Unacked Archive DB.
#Size in multiple of Req_Buffer_Length
#A value of 0 indicates no max value.
#Range : 1 - 1000
#Default: 10
Max_Storage_Count = 10
```

```
#Key: Vacuumize_DB
#Vacuumize database periodically.
#Value : yes / no
```

```
#Default: yes (Vacuumize database periodcally)
Vacuumize_DB = yes

#Section: DB_Param
#Contains config values for db parameters.
[DB_Param]
#Key: DB_Host_Name
#DB server name or address string
#Default : No default
DB_Host_Name = 127.0.0.1

#Key: DB_User_Name
#User name for connection with database
#Value : User name string
#Default: postgres
#Description : PostgreSQL login
DB_User_Name = gsspg

#Key: DB_Password
#Password for connection with database
#Value : User name string
#Default: postgres
#Description :PostgreSQL passwd
DB_Password = gsspg

#Key: DB_Host_Addr
#DB server address.
#Default : No Default, if given overrides DB_Host_Name
DB_Host_Addr =
```

```
#Key: DB_Port

#DB server udp port value
#Range : 1 - 65535
#Default: 5432
#Description :PostgreSQL port
DB_Port = 5499

#Key: DB_Retry_Count
#Retry count to perform reconnection with DB server
#Range : 10 - 100
#Default: 20
DB_Retry_Count = 20

#Key: DB_Retry_Interval
#Retry Interval (secs) to perform reconnection with DB server
#Range : 1 - 10 secs
#Default: 2 secs
DB_Retry_Interval = 2

#Section: Notif_Param
#Contains config values for resource notification parameters.

[Notif_Param]

#Key: Notif_Type
#Value : Alarm / Alert

#Alarm : Generates notification once for each threshold cross over for the
monitored entity.

#Alert : Generates notification on each polling interval for the monitored
entity only if above threshold mark.

#Default: Alarm
```

```
Notif_Type = Alarm
```

```
#Key: Notif_Poll_Interval
```

```
#Notification poll interval (mins) to examine the state of the monitored entity.
```

```
#Range : 1 - 1440 mins
```

```
#If threshold configuration is disabled then default: 0 (Disable: No notification)
```

```
#If threshold configuration is enabled then default: 1
```

```
Notif_Poll_Interval = 1
```

```
#Key: Notif_Retry_Interval
```

```
#Notification retry interval (secs)
```

```
#Range : 1 - 60 secs
```

```
#Default: 20 secs
```

```
Notif_Retry_Interval = 20
```

```
#Key:Notif_Retry_Count
```

```
#Notification retry count
```

```
#Range : 1 - 5
```

```
#Default: 3
```

```
Notif_Retry_Count = 3
```

```
#Key: Notif_GCDR_File_Period
```

```
#Enables Alarm / Alert to be sent for files older than given period in mins
```

```
#Note : Even though name of this parameter contains GCDR, this parameter is equally applicable to all type of CDRs supported by GSS
```

```
#Range : 1 - 7*24*60 mins
```

```
#Default: 15
```

```
Notif_GCDR_File_Period = 15
```

```
#Key: Notif_CPU_Usage
#Enables Alarm / Alert to be sent for system cpu usage in %.
#Range : 0 - 100 %
#Default: 60
Notif_CPU_Usage = 60

#Key: Notif_Mem_Usage
#Enables Alarm / Alert to be sent for system swap usage in MB.
#Range : 0 - ULONG_MAX (MB)
#Default: 9500
Notif_Mem_Usage = 9500

#Key: Enable_Cluster_Alarms
#Flag to indicate whether to enable cluster notifications.
#Value : yes/no
#Default: no
Enable_Cluster_Alarms = y

#Key: Cluster_Path
#Path to installed cluster components.
#This has to be valid path.
#Value : Cluster path string
#Default: /usr/cluster/bin
Cluster_Path = /usr/cluster/bin

Key: Cluster_Name
#Cluster name string set during installation. Please do not edit this field.
#Default : No default
Cluster_Name = GSS_HA
```

```
#Key: Cluster_Node_List

#Cluster nodes list string set during installation. Please do not edit this
field.

#Default : No default

Cluster_Node_List = clustems2,clustems1

#Key: Cluster_Node_Count

#Cluster node count.

#Default : 0

Cluster_Node_Count = 2

#Key: Notif_Disk_Usage

#Enables Alarm / Alert to be sent

#if available disk size(GB) falls below configured limit.

#Range : 0 - ULONG_MAX (GB)

#Default: 199

Notif_Disk_Usage = 199

#Section: Log_Param

#Contains config values for logging and status.

[Log_Param]

#Key Show_Status

#Show status periodically (secs)

#Range : 0 - ULONG_MAX secs

#Default: 300 secs

#0 - Disables

Show_Status = 300
```

```
#Key: Print_IO_Msg
#Prints I/O messages to log
#Value : yes /no
#Default: no
Print_IO_Msg = no
```

```
#Key: Audit_File_Size
#Audit log file size
#minimum 2500 kbytes
#default 2500 kbytes
Audit_File_Size = 2500
```

# Troubleshooting the GSS

This section provides some basic troubleshooting tips that can be referred to in the event of any failure of the GSS application.

## Problems During Installation

During the installation process, if you receive error messages such as:

```
syntax error near unexpected token
```

**Step 1** Check the version of the OS by entering the following command:

```
uname - a
```

**Step 2** Check the version of the bash by entering the following command:

```
/bin/bash -version
```

The OS must be greater than or equal to Solaris 9 and the bash version should be 3 or higher.

If you do not have the correct server operating system software, contact your sales representative.

## Problems Managing a Cluster Node

These following commands can only be issued from the console on a cluster node and the user must have root privileges.

Cluster console commands are accessed from the `/<clus_install_dir>/gss` directory.

### Check Status of Node in Cluster

There are several commands that can confirm node status. The first lists the resource group, the nodes within the group and the status of the nodes. The second command provides a more detailed display of the cluster resource group.

```
clresourcegroup status
```

```
scstat
```

### Change to Stand-alone Mode - Cluster Console Command

This command **MUST** be entered from the console. It terminates GSS operations and switches the node from Cluster mode to Stand-alone mode:

```
shutdown
```

At the 'ok' prompt, enter:

```
boot -x
```

## GSS/FileGen/PostgreSQL Not Running

Begin by confirming what processes are running on the server.

### GSS/FileGen Status

```
ps -ef | grep gss
```

Minimally, the resulting display should include information similar to the following:

```
0 15052 07:23:25 TS 59 0:00 /<install_dir>/gss/bin/gssfilegen
0 15062 07:23:25 TS 59 0:00 /<install_dir>/gss/bin/gss
```

The above information indicates that both GSS and FileGen are running.

If you do not receive similar information, then you need to **start** GSS which will automatically start GSS, FileGen, PostgreSQL, and PSMON.

- **Stand-alone Node** - enter the following command:

```
cd /<install_dir>/gss
./GSS start
```

If GSS fails to start, then contact your sales representative.

- **GSS Node in Cluster** - enter the following command:

```
cd /<install_dir>/gss
clresourcegroup manage <Resource Group>
clresourcegroup online -n <nodename> -eM <Resource Group>
```

If GSS fails to start, then contact your sales representative.

### PostgreSQL Status

```
ps -ef | grep postgres
```

Minimally, the resulting display should include information similar to the following:

```
postgres 15080 14972 0 07:23:25 ? 0:00 /<install_dir>/ postgres/bin/postgres -D
```

The above information indicates that PostgreSQL is running.

- **Stand-alone Node** - If you do not receive similar information, then you need to restart the GSS stand-alone node which will automatically start the GSS, FileGen, PostgreSQL, and PSMON. To restart GSS on stand-alone node, refer to [Accessing GSS Commands](#) section.

- **GSS Node in Cluster** - If you do not receive similar information, then you need to restart the GSS node which will automatically stop all processes and then start the GSS, FileGen, PostgreSQL, and PSMON. To restart GSS on a cluster node, refer to [Accessing GSS Commands](#) section.

## Support Detail Logs for Debugging

If you encounter problems with GSS and need to seek assistance from customer support, you may be asked to send them support detail logs which are generated with the *SupportDetail* utility script. This script runs system commands and captures the output and resulting logs in an XML file.

The *SupportDetails* utility is found in the `<gss_install_dir>/gss/tools/supportdetails` directory. There is also a README file that provides details about the use of the script.

**Step 1** From the root directory, change to the `<gss_install_dir>/gss/tools/supportdetails` directory.

**Step 2** Check the command syntax:

```
./getSupportDetails.pl --help
```

**Step 3** Run the script:

```
./getSupportDetails.pl --level=<number>
```

`<number>` specifies the level of debug actions to run. There are four levels:

- Level 1 generates a simple status log and collects the GSS configuration file.
- Level 2 collects Level 1 information plus
  - any available database logs
  - information about current the Solaris version and installed patches
  - information about the hard disk partitioning
- Level 3 collects Level 1 and 2 information plus the system configuration file and messages files stored in `/var/adm`.
- Level 4 is the most comprehensive and includes all Level 1, 2, and 3 information and all logs of the current node, plus
  - information about all packages installed
  - traces of any crash files (if debugger installed)
  - output of the following commands: `ipcs, ps -eaf, scstat, prtdiag`

The README file provides a complete description of the levels.

The script generates a zipped file, `gsssupportDetails.tar.gz` and puts it in `/tmp/log` directory. You can e-mail this gzip to customer support for debugging purposes. This information will provide them with a quick but fairly complete view of the status of your system and the configuration.

# Chapter 4

## Managing the GSN-GSS Services

---

This chapter provides the procedures for setting up GSS support on a ASR 5000 GSN - either an SGSN or a GGSN.

To edit the GSN configuration, you must be connected to the GSN using the standard chassis' command line interface (CLI). The recommended access technique is via the console port on the Switch Processor I/O (SPIO) card. CLI access and use information, beyond that provided in this chapter, is available in the Command Line Interface Reference.

Procedures mentioned in this chapter must be performed at the command line of the GSN that is to communicate with a GTPP Storage Server (GSS).

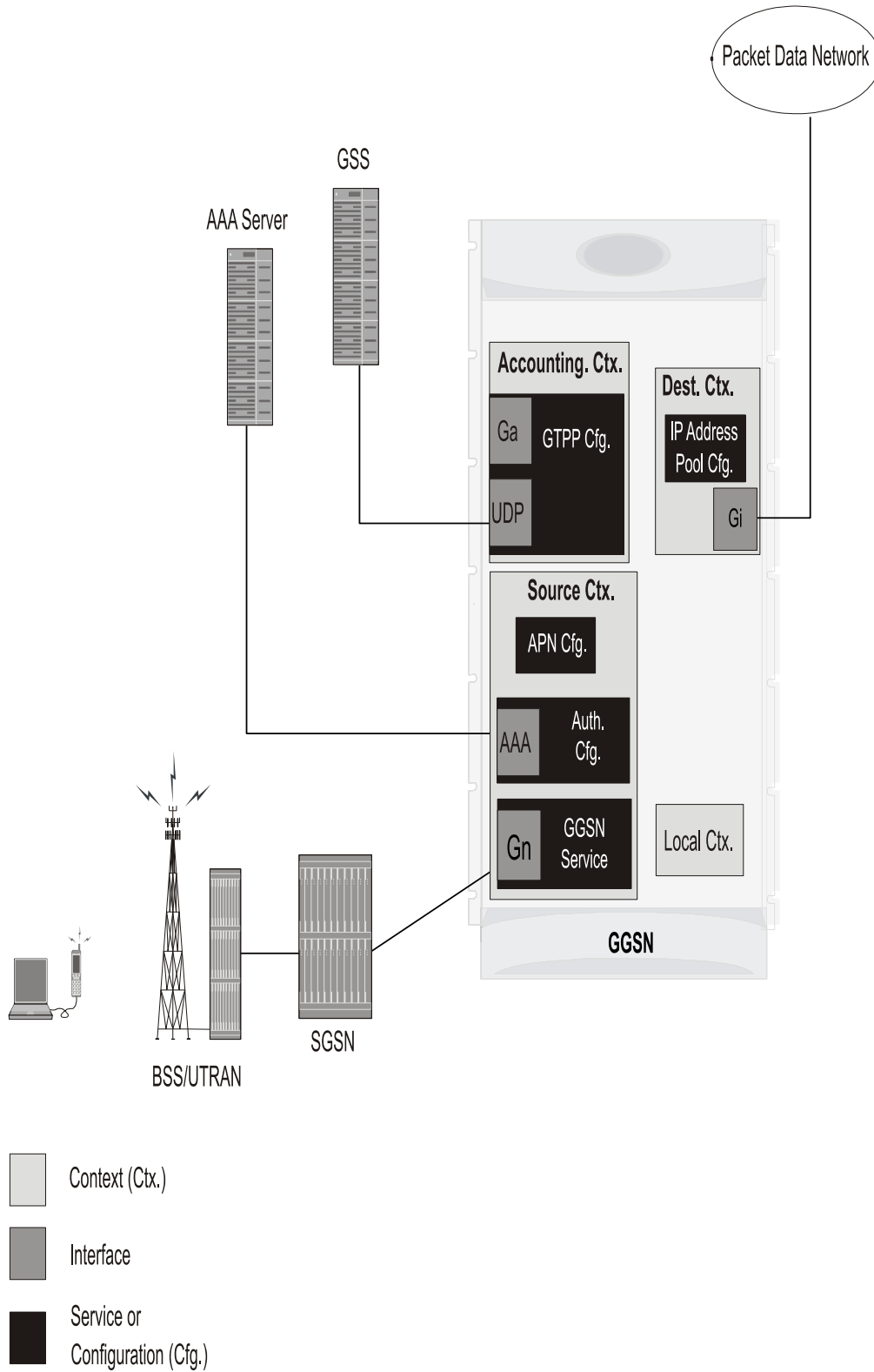
- [Configuring GGSN Support for GSS](#)
- [Configuring SGSN Support for GSS](#)

## Deployment of GSS with a GSN

This section provides information about the deployment of a gateway GPRS support node (GGSN) connecting to a GTPP storage server (GSS). For more information about the GSS deployment scenarios, refer to the *GTPP Storage Server Overview* chapter in this guide.

The following figure shows a typical example of a GSS deployment with a GSS connected to a GGSN.

Figure 5. GSS in the GPRS/UMTS Network



The logical source context on the GGSN facilitates:


- GPRS Tunneling Protocol Prime (GTPP) configuration of the Ga interface
- UDP interface to the GSS
- Optional Network-requested PDP context processing
- Access Point Name (APN) configuration
- RADIUS authentication configuration (Auth.cfg) and the interface (AAA) to the authentication server
- GGSN service(s) and Gn interface to the Service GPRS Support Node (SGSN)

The logical destination context on the GGSN facilitates:

- IP address pools
- Gi interface to the packet data network (PDN)

## Configuring GGSN Support for GSS

---


 **Important:** Before proceeding, the GGSN must be configured for all related services, such as the GGSN service, and the GTPP accounting support which can include support for GGSN use of a CGF. For more information on the GGSN configurations and procedures, refer to the *GGSN Configuration Example* chapter and the *Service Configuration Procedures* chapter in the *GGSN Administration Guide*.

---

In order to support connectivity with a GSS, the GGSN system is configured with two new components:

- GTPP Storage Server is configured in the same source context as the GGSN service(s) or any other accounting context.
- UDP interface is bound to the GTPP Storage Server. The UDP interface is a proprietary interface used by the GGSN system to communicate with the GSS.

---

 **Important:** GSS support on the GGSN can be configured in any context other than the local context. It is recommended that GSS component support be configured in the same source context as the GGSN service(s) - see Figure 5 above.

---

These instructions assume that you are connected to the GGSN and are using the GGSN's command line interface (CLI) beginning at the root prompt in the Exec mode:

```
[local]host_name#
```

**Step 1** Enter the configuration mode by entering the following command:

```
configure
```

The following prompt appears:

```
[local]host_name(config)#
```


**Step 2** Enter the following command to configure the system to reserve a CPU for performing a proxy function for accounting:

```
gtp single-source [ private-extensions ] [ centralized-lsrn-creation ]
```

*private-extensions* is an optional keyword and enables the proprietary use of the customer-specific GTPP extensions.

*centralized-lsrn-creation* optional keyword defines the LSRN generation at proxy. With this option enabled, the eG-CDRs will skip the session manager and be sent to the AAA manager and the AAA proxy will generate the LRSN for both the G-CDRs and the eG-CDRs.

---

 **Important:** If *private-extensions* is not configured, all the customer specific private extensions related to the GTPP message transfer with the CGF and recovery through the GSS are disabled. Additionally, this command is customer specific. Please contact your local sales representative for additional information. For detailed information, see the **gtp** **single-source** command in the *Global Configuration Mode* chapter of the *Command Line Interface Reference*.

---

**Step 3** Enter the context configuration mode by entering the following command:

```
context context_name
```

*context\_name* is the name of the system source context designated for GTPP accounting configuration. The name must be from 1 to 79 alpha and/or numeric characters and is case sensitive.

The following prompt appears:

```
[<context_name>]host_name(config-ctx)#
```

**Step 4** Configure the IP address of the system interface within the current context used to communicate with the CGF. This command establishes a Ga interface for the system - one per gtp group and there can be multiple gtp groups. The interface is configured by entering the following command:

```
gtp charging-agent address ip_address [port port]
```

*ip\_address* specifies the IP address of the interface configured within the current context that is used to transmit CDRs to the CGF. It must be configured using dotted decimal notation.

*port* specifies the charging agent UDP port and must be followed by an integer, ranging from 1 to 65535. Default port number is 49999.

Multiple interfaces can be configured within a single context if needed.

---

 **Important:** Configuring the GTPP charging-agent on port 3386 may interfere with GGSN-service configured with the same IP address.

---

**Step 5** Optional. Configure the charging gateway function (CGF) accounting server(s) that the GGSN system is to communicate with by entering the following command:

```
gtp server ip_address [max msgs] [priority priority] [udp-port port] [node-alive { enable | disable }]
```

Keyword/Variable	Description
ip_address	Specifies the IP address of the CGF.
max	Specifies the maximum number of outstanding or unacknowledged GTPP packets (from any one AAA Manager task) allowed for this CGF. <i>msgs</i> can be configured to any integer value from 1 to 256. The default is 256.

Keyword/Variable	Description
priority	Specifies the relative priority of this CGF. When multiple CGFs are configured, the priority is used to determine which CGF server to send accounting data to. <i>priority</i> can be configured to any integer value from 1 to 1000. The default is 1. Lower values are recommended.
udp-port	Specifies the UDP port over which the GGSN communicates with the CGF. <i>port</i> can be configured to any integer value between 1 and 65535. The default is 3386.
node-alive { enable   disable }	Default: Disable. This optional keyword allows operator to enable/disable GGSN to send Node Alive Request to GTPP Server (i.e. CGF). This configuration can be done per GTPP Server basis.

Multiple servers can be configured within a single context if needed. Repeat this step as needed to configure connections with multiple CGFs.

**Step 6** Define the GTPP storage server that the GGSN will use to store CDRs by entering the following command:

```
gtp storage-server ip-address port port_num
```

*ip\_address* specifies the IP address of the GSS and *port\_num* specifies the UDP port over which the GGSN communicates with the GSS. *port* can be configured to any integer value between 1 and 65535.



**Important:** The `gtp storage-server` command only takes affect if the `gtp single-source [private-extensions]` in the Global Configuration Mode has also been configured in *step 2* of this procedure. For more details, check these command descriptions in the *Command Line Interface*. Additionally, this command is customer specific and you may need to contact your local sales representative for additional information.

**Step 7** Configure the GGSN CDR transmission to maintain peak GSS performance by setting the maximum number of CDRs per packet and the wait timer for packet transmission. Use the following command with recommended settings of 255 CDRs and 5 seconds wait:

```
gtp max-cdrs 255 wait time 5
```

**Step 8** Optional. Configure the maximum number of retries to communicate with a GTPP storage server before putting that server as dead by entering the following command:

```
gtp storage-server max-retries max_attempts
```

*max\_attempts* specifies the number of times the system attempts to communicate with a GTPP back-up storage server that is not responding and it can be configured to any integer.

**Step 9** Optional. Configure the timeout duration to wait between two attempts by entering the following command:

```
gtp storage-server timeout duration
```

*duration* specifies the maximum amount of time the system waits for a response from the GTPP back-up storage server before assuming the packet is lost. *duration* is measured in seconds and can be configured to any integer value from 30 to 120.

**Step 10** Repeat *step 3* through *step 8* as needed to configure multiple GSSs with or without CGF.


**Step 11** Optional. Configure the GTPP dictionary with predefined CDR attributes by entering the following command:

```
gtpp dictionary { custom1 | custom10 | custom11 | custom12 | custom13 | custom14
| custom15 | custom16 | custom17 | custom18 | custom19 | custom2 | custom20 |
custom3 | custom4 | custom5 | custom6 | custom7 | custom8 | custom9 | standard }
```

For information on the GTPP dictionaries, refer to the *G-CDR and Enhanced G-CDR Field Reference Tables* chapter of the *AAA and GTPP Interface Administration and Reference*.

**Step 12** Optional. Configure optional fields for CDRs by entering the following command:

```
gtpp attribute { diagnostics | duration-ms | node-id-suffix STRING | local-
record-sequence-number | plmn-id }
```

Keyword/Variable	Description
diagnostics	<p>Includes the Diagnostic field in the CDR that is created when PDP contexts are released. The field will contain one of the following values:</p> <ul style="list-style-type: none"> <li>• 36 - if the SGSN sends us "delete PDP context request".</li> <li>• 38 - if the GGSN sends "delete PDP context request" due to GTP-C/GTP-U echo timeout with SGSN.</li> <li>• 40 - if the GGSN sends "delete PDP context request" due to receiving a RADIUS Disconnect-Request message.</li> <li>• 26 - if the GGSN sends "delete PDP context request" for any other reason (e.g., the operator types "clear subscribers" on the GGSN).</li> </ul> <p>This disabled by default.</p>
duration-ms	<p>Specifies that the information contained in the mandatory Duration field be reported in milliseconds instead of seconds as the standards require.</p> <p>This disabled by default.</p>
node-id-suffix STRING	<p>Default: Disabled</p> <p>Specifies the string suffix to use in the NodeID field of GTPP G-CDRs. Each Session Manager task generates a unique NodeID string per GTPP context.</p> <p><i>STRING</i>: This is the configured Node-ID-Suffix having any string between 1 to 16 characters.</p> <p>The NodeID field is a printable string of the <i>ndddSTRING</i> format:</p> <p><i>n</i>: The first digit is the Session manager restart counter having a value between 0..7.</p> <p><i>ddd</i>: The number of sessmgr instances.</p> <p>Uses the specified NodeID-suffix in all CDRs. The "Node-ID" field consists of sessMgr Recovery counter (1 digit) <i>n</i> + AAA Manager identifier (3 digits) <i>ddd</i> + the configured Node-Id-suffix (1 to 16 characters) <i>STRING</i>.</p> <hr/> <p> <b>Important:</b> If this <b>node-id-suffix</b> is not configured, the GGSN uses the GTPP context name as the Node-id-suffix (truncated to 16 characters).</p> <hr/>
local-record- sequence-number	<p>Includes the Node ID field in the CDR that is created when PDP contexts are released. The field will consist of a AAA Manager identifier automatically appended to the name of the GGSN service.</p> <p>The name of the GGSN service may be truncated, because the maximum length of the Node ID field is 20 bytes. Since each AAA Manager generates CDRs independently, this allows the Local Record Sequence Number and Node ID fields to uniquely identify a CDR.</p> <p>This disabled by default.</p>

Keyword/Variable	Description
plmn-id	Reports the RAI as the SGSN PLMN Identifier value if it was originally provided by the SGSN in the GTP create PDP context request. This is enabled by default.

**Step 13** Return to the root prompt by entering the following command:

```
end
```

The following prompt appears:

```
[local]host_name#
```

**Step 14** Verify that the GSS parameters were configured properly by entering the following command:

```
show configuration | grep gtp
```

**Step 15** Optional. If CGF is configured, verify that the CGF parameters were configured properly by entering the following command:

```
show gtp accounting servers
```

**Step 16** Verify that the GTP storage server parameters, statistics and status by entering the following command:

```
show gtp storage-server [ counters | statistics | status [ verbose ] ] [ {  
grepgrep_options | more } ] | }
```



**Important:** For more commands and keywords to configure GTPP, CGF and related feature, refer to the *Command Line Interface Reference* and the *GSN Configuration Example* chapter in the *GSN System Administration Guide*.

**Step 17** Save this modified GSN system configuration as described in the *Saving the GSN Configuration* section in this chapter.

## Configuring SGSN Support for GSS



**Important:** Before proceeding, the SGSN must be configured for all related services, such as the SGSN service, and the GTPP accounting support which can include support for the SGSN use of a CGF. For more information on the SGSN configurations and procedures, refer to the *GSN Configuration Example* chapter and the *Service Configuration Procedures* chapter in the *GSN Administration Guide*.

In order to support connectivity with a GSS, the SGSN system is configured with two new components:

- GTPP Storage Server can be configured in any context but it is recommended to be configured in the accounting context in the SGSN service.
- Ga interface is bound to the accounting context of the GTPP Storage Server. The Ga interface is used by the SGSN system to communicate with the GSS.



**Important:** GSS support on the SGSN can be configured in any context other than the local context. It is recommended that GSS component support be configured in the accounting context .

These instructions assume that you are connected to the SGSN and are using the SGSN's command line interface (CLI) beginning at the root prompt in the Exec mode:

```
[local]host_name#
```

**Step 1** Enter the configuration mode by entering the following command:

```
configure
```

The following prompt appears:

```
[local]host_name(config)#
```

**Step 2** Enter the following command to configure the system to reserve a CPU for performing a proxy function for accounting:

```
gtpp single-source [ private-extensions ] [ centralized-lsrn-creation ]
```

*private-extensions* is an optional keyword and enables the proprietary use of the customer-specific GTPP extensions.

*centralized-lsrn-creation* optional keyword defines the LSRN generation at proxy. With this option enabled, the S-CDRs or M-CDRs will skip the session manager and be sent to the If *private-extensions* is not configured, all the customer specific private extensions related to the GTPP message transfer with the CGF and recovery through the GSS are disabled. Additionally, this command is customer specific. Please contact your local sales representative for additional information. For detailed information, see the **gtpp single-source** command in the *Global Configuration Mode* chapter of the *Command Line Interface Reference*.

**Step 3** Enter the context configuration mode by entering the following command:

```
context accounting_context
```

The context name must be from 1 to 79 alpha and/or numeric characters and it is case sensitive.

The following prompt appears:

```
[accounting_context]host_name(config-ctx)#
```

**Step 4** Configure the IP address of the system interface within the current context used to communicate with the CGF. This command establishes a Ga interface for the system - one per gtpp group and there can be multiple gtpp groups. The interface is configured by entering the following command:

```
gtpp charging-agent address ip_address [port port]
```

*ip\_address* specifies the IP address of the interface configured within the current context that is used to transmit CDRs to the CGF. It must be configured using dotted decimal notation.

*port* specifies the charging agent UDP port and must be followed by an integer, ranging from 1 to 65535. Default port number is 49999.

Multiple interfaces can be configured within a single context if needed.



**Important:** Configuring the GTPP charging-agent on port 3386 may interfere with SGSN-service configured with the same IP address.

**Step 5** Optional. Configure the charging gateway function (CGF) accounting server(s) that the SGSN system is to communicate with by entering the following command:

```
gtpp server ip_address [ max msgs ] [ priority priority ] [ udp-port port ] [ node-alive {
enable | disable } ]
```

Keyword/Variable	Description
ip_address	Specifies the IP address of the CGF.
max	Specifies the maximum number of outstanding or unacknowledged GTPP packets (from any one AAA Manager task) allowed for this CGF. <i>msgs</i> can be configured to any integer value from 1 to 256. The default is 256.
priority	Specifies the relative priority of this CGF. When multiple CGFs are configured, the priority is used to determine which CGF server to send accounting data to. <i>priority</i> can be configured to any integer value from 1 to 1000. The default is 1. Lower values are recommended.
udp-port	Specifies the UDP port over which the SGSN communicates with the CGF. <i>port</i> can be configured to any integer value between 1 and 65535. The default is 3386.
node-alive { enable   disable }	Default: Disable. This optional keyword allows operator to enable/disable SGSN to send Node Alive Request to GTPP Server (i.e. CGF). This configuration can be done per GTPP Server basis.


Multiple servers can be configured within a single context if needed. Repeat this step as needed to configure connections with multiple CGFs.

**Step 6** Define the GTPP storage server (GSS) that the SGSN will use to store CDRs by entering the following command:

```
gtpp storage-server ip-address port port_num
```

*ip\_address* specifies the IP address of the GSS and *port\_num* specifies the UDP port over which the SGSN communicates with the GSS. *port* can be configured to any integer value between 1 and 65535.

---

 **Important:** The `gtpp storage-server` command only takes affect if the `gtpp single-source [private-extensions]` in the Global Configuration Mode has also been configured in *step 2* of this procedure. For more details, check these command descriptions in the *Command Line Interface*. Additionally, this command is customer specific and you may need to contact your local sales representative for additional information.

---

**Step 7** Configure the SGSN CDR transmission to maintain peak GSS performance by setting the maximum number of CDRs per packet and the wait timer for packet transmission. Use the following command with recommended settings of 255 CDRs and 5 seconds wait:

```
gtpp max-cdrs 255 wait time 5
```

**Step 8** Optional. Configure the maximum number of retries to communicate with a GTPP storage server before putting that server as dead by entering the following command:

```
gtpp storage-server max-retries max_attempts
```

*max\_attempts* specifies the number of times the system attempts to communicate with a GTPP back-up storage server that is not responding and it can be configured to any integer.

**Step 9** Optional. Configure the timeout duration to wait between two attempts by entering the following command:

**gtp storage-server timeout duration**

*duration* specifies the maximum amount of time the system waits for a response from the GTPP back-up storage server before assuming the packet is lost. *duration* is measured in seconds and can be configured to any integer value from 30 to 120.

**Step 10** Repeat *step 3* through *step 8* as needed to configure multiple GSSs with or without CGF.


**Step 11** Optional. Configure the GTPP dictionary with predefined CDR attributes by entering following command: (Currently, it is recommended that either custom 8 or custom 10 for an SGSN.)

```
gtp dictionary { custom1 | custom10 | custom11 | custom12 | custom13 | custom14
| custom15 | custom16 | custom17 | custom18 | custom19 | custom2 | custom20 |
custom3 | custom4 | custom5 | custom6 | custom7 | custom8 | custom9 | standard }
```

For information on the GTPP dictionaries, refer to the *SGSN and Mobility Management Charging Detail Record Field Reference Tables* chapter of the *AAA Interface Administration and Reference*.

**Step 12** Optional. Configure optional fields for CDRs by entering the following command:

```
gtp attribute { diagnostics | duration-ms | node-id-suffix STRING | local-
record-sequence-number | plmn-id }
```

Keyword/Variable	Description
diagnostics	Includes the Diagnostic field in the CDR that is created when PDP contexts are released. The field will contain one of the following values: <ul style="list-style-type: none"> <li>• 36 - if the SGSN sends us "delete PDP context request".</li> <li>• 38 - if the GGSN sends "delete PDP context request" due to GTP-C/GTP-U echo timeout with SGSN.</li> <li>• 40 - if the GGSN sends "delete PDP context request" due to receiving a RADIUS Disconnect-Request message.</li> <li>• 26 - if the GGSN sends "delete PDP context request" for any other reason (e.g., the operator types "clear subscribers" on the GGSN).</li> </ul> This disabled by default.
duration-ms	Specifies that the information contained in the mandatory Duration field be reported in milliseconds instead of seconds as the standards require. This disabled by default.
node-id-suffix STRING	Default: Disabled Specifies the string suffix to use in the NodeID field of GTPP CDRs. Each Session Manager task generates a unique NodeID string per GTPP context. <i>STRING</i> : This is the configured Node-ID-Suffix having any string between 1 to 16 characters. The NodeID field is a printable string of the <i>ndddSTRING</i> format: <i>n</i> : The first digit is the Session manager restart counter having a value between 0..7. <i>ddd</i> : The number of sessmgr instances. Uses the specified NodeID-suffix in all CDRs. The "Node-ID" field is consists of sessMgr Recovery counter (1 digit) <i>n</i> + AAA Manager identifier (3 digits) <i>ddd</i> + the configured Node-Id-suffix (1 to 16 characters) <i>STRING</i> .   <b>Important:</b> If this <i>node-id-suffix</i> is not configured, the GGSN uses the GTPP context name as the Node-id-suffix (truncated to 16 characters).

Keyword/Variable	Description
plmn-id	Reports the RAI as the SGSN PLMN Identifier value if it was originally provided by the SGSN in the GTP create PDP context request. This is enabled by default.

**Step 13** Return to the root prompt by entering the following command:

```
end
```

The following prompt appears:

```
[local]host_name#
```

**Step 14** Verify that the GSS parameters were configured properly by entering the following command:

```
show configuration | grep gtp
```

**Step 15** Optional. If CGF is configured, verify that the CGF parameters were configured properly by entering the following command:

```
show gtp accounting servers
```

**Step 16** Verify that the GTP storage server parameters, statistics and status by entering the following command:

```
show gtp storage-server [ counters | statistics | status [ verbose ] ] [ | {  
grep grep_options | more } ]
```



**Important:** For more commands and keywords to configure GTP, CGF and related feature, refer to the *Command Line Interface Reference* and the *SGSN Configuration Example* chapter in the *SGSN System Administration Guide*.

**Step 17** Save this modified SGSN system configuration as described in the *Saving the GSN Configuration* section in this chapter.

## Saving the GSN Configuration

After modifying the GSN system configuration to include support for GSN-GSS services, the GSN configuration information can be saved to a file locally (on the GSN) or to a remote node on the network. This file is known as a configuration file and can also be used on any other systems that require the same configuration.


Files saved locally can be stored in the CompactFlash on the SPC/SMC or on a PCMCIA memory card installed in either slot 1 or 2 of the SPC/SMC. Files that are saved to a remote network node can be transmitted using either FTP or TFTP.





These instructions assume that you are at the prompt for the Exec mode in the chassis' CLI -

```
[local]host_name#
```

To save your configuration, enter the following command:

```
save configuration url [-redundant] [-noconfirm] [showsecrets] [verbose]
```

Keyword/Variable	Description
url	<p>Specifies the path and name to which the configuration file is to be stored. url may refer to a local or a remote file. url must be entered using one of the following formats:</p> <ul style="list-style-type: none"> <li>• {/flash   /pcmcia1   /hd} [/directory] /file_name</li> <li>• file:/{/flash   /pcmcia1   /hd} [/directory] /file_name</li> <li>• tftp://{ipaddress   host_name [:port#]} [/directory] /file_name</li> <li>• ftp://[username [:password] @] {ipaddress   host_name} [:port#] [/directory] /file_name</li> <li>• sftp://[username [:password] @] {ipaddress   host_name} [:port#] [/directory] /file_name</li> </ul> <p>/flash corresponds to the CompactFlash on the SPC/SMC.                      /pcmcia1 corresponds to PCMCIA slot 1.                      /pcmcia2 corresponds to PCMCIA slot 2.                      ipaddress is the IP address of the network server.                      host_name is the network server's hostname.</p> <hr/> <p> <b>Important:</b> Use of the SMC hard drive as an option is not supported in this release.</p>

Keyword/Variable	Description
	<p><i>port#</i> is the network server's logical port number. Defaults are:</p> <ul style="list-style-type: none"> <li>• tftp: 69 - data</li> <li>• ftp: 20 - data, 21 - control</li> <li>• sftp: 115 - data</li> </ul> <hr/> <p> <b>Important:</b> <i>host_name</i> can only be used if the <b>networkconfig</b> parameter is configured for DHCP and the DHCP server returns a valid nameserver.dx.</p> <hr/> <p><i>username</i> is the username required to gain access to the server if necessary.  <i>password</i> is the password for the specified username if required.  <i>/directory</i> specifies the directory where the file is located if one exists.  <i>/file_name</i> specifies the name of the configuration file to be saved.</p> <hr/> <p> <b>Important:</b> Configuration files should be named with a .cfg extension.</p>
-redundant	<p>Optional: This keyword directs the system to save the CLI configuration file to the local device, defined by the url variable, and then automatically copy that same file to the like device on the standby SPC or SMC, if available.</p> <hr/> <p> <b>Important:</b> This keyword will only work for local devices that are located on both the active and standby SPCs/SMCs. For example, if you save the file to the /pcmcia1 device on the active SPC, that same type of device (a PC-Card in Slot 1 of the standby SPC) must be available. Otherwise, a failure message is displayed. If saving the file to an external network (non-local) device, the system disregards this keyword.</p> <hr/> <p> <b>Important:</b> This keyword does not synchronize the local file system. If you have added, modified, or deleted other files or directories to or from a local device for the active SPC/SMC, then you must synchronize the local file system on both SPCs/SMCs.</p>
-noconfirm	Optional: Indicates that no confirmation is to be given prior to saving the configuration information to the specified filename (if one was specified) or to the currently active configuration file (if none was specified).
showsecrets	Optional: This keyword causes the CLI configuration file to be saved with all passwords in plain text, rather than their default encrypted format.
verbose	Optional: Specifies that every parameter that is being saved to the new configuration file should be displayed.

To save a configuration file called *system.cfg* to a directory that was previously created on the SPC/SMC CompactFlash called *cfgfiles*, enter the following command:

```
save configuration /flash/cfgfiles/system.cfg
```

To save a configuration file called *simple\_ip.cfg* to a directory called *host\_name\_configs* using an FTP server with an IP address of 192.168.34.156 on which you have an account *administrator* and password *secure*, enter the following command:

**save configuration****ftp://administrator:secure@192.168.34.156/host\_name\_configs/simple\_ip.cfg**

To save a configuration file called *init\_config.cfg* to the root directory of a TFTP server with a hostname of *config\_server*, enter the following command:

**save configuration tftp://config\_server/init.config.cfg**