Provisioning the Cisco HSI

Introduction

This chapter describes the provisionable data requirements for the Cisco H.323 Signaling Interface (HSI). The data is divided into two areas: system configuration and H.323 stack data. This chapter contains the following sections:

- Cisco HSI Configuration, page 5-1
- H.323 Stack Configuration, page 5-6

Cisco HSI Configuration

All configuration data is contained within configuration files. Cisco HSI starts with an initial configuration file in $BASEDIR/currentGW/etc/GWmain.conf. This file is created during installation of the software.

The configuration data within the file is defined as dynamic, static, or constant:

- Dynamic data can be modified by a provisioning session (see Appendix A, “MML Commands”). It can be activated on the currently running Cisco HSI.
- Static data can be modified by a provisioning session but cannot be activated on a running Cisco HSI. Changes to dynamic and static data can be written to a separate provisioning file (in $BASEDIR/currentGW/var/prov/configname/session.dat) that can be used during subsequent restarts of the Cisco HSI.
- Constant configuration data is contained within the configuration file and cannot be modified by provisioning sessions. Constant configuration data can be modified only by system technicians or administrators who use UNIX editing tools. This data is replicated from the initial configuration file into the provisioning files, and between subsequent provisioning sessions.

  Examples of the use of constant data are given in Appendixes C, D, E, and F. These appendixes determine the mapping of cause values for incoming and outgoing H.323 and E-ISUP messages. System technicians can modify these values in the initial configuration file to explicitly choose the mappings for their system.

When a provisioning session creates a new configuration file, it also verifies that provisioned data is within allowable ranges and indicates this in the start of the file. It checksums the file and writes the checksum as $BASEDIR/currentGW/var/prov/configname/checksum.dat. When the Cisco HSI starts up, it attempts to read the active configuration, checks that the configuration has been verified, and ensures that the checksum matches. If the active configuration is not verified or if the checksum is faulty, the configuration reverts to using the $BASEDIR/currentGW/etc/GWmain.conf file.

All configuration data that can be set in the system is defined in the Skeleton Configuration file (see Appendix B, "Skeleton Configuration File"). The Skeleton Configuration file defines the data names and types (strings or numbers), and defines whether the data is dynamic, static, or constant. Flexibility in changing data types to allow or deny provisioning is therefore provided.

MML Configuration Commands

The three types of MML configuration commands are as follows:

- Configuration session commands that work with entire provisioning data files (see Table 5-1)
- Configuration component or parameter commands that perform actions on components or parameters affecting a specific data file (see Table 5-2)
- Configuration export commands

For more information about MML configuration commands, see Appendix A, "MML Commands."

Note Parameter names are not case sensitive.

Table 5-1 Configuration Session Commands

Command	Description
prov-sta	Starts a provisioning session to create a new configuration or modify an existing configuration
prov-cpy	Activates the configuration settings in the current provisioning session
prov-stp	Terminates the provisioning session and saves the configuration

Table 5-2 Configuration Component or Parameter Commands

Command	Description
prov-add	Adds a component to the Cisco HSI
prov-dlt	Deletes a provisioned component
prov-ed	Modifies a provisioned component
prov-rtrv	Retrieves information about an existing provisioning session

The configuration export command is the **prov-exp** command, which exports the current provisioned configuration of the Cisco HSI in MML command form to a file.

System Configuration Data

System configuration data can be static or dynamic. Static data can be activated only at startup. Dynamic data can be activated during system run time.

Static System Data

To modify the static system data parameters in Table 5-3, use the **sys_config_static** MML name variable for the **prov-add**, **prov-dlt**, and **prov-ed** commands. Halt and restart the application for the changes to take effect.

The parameters in Table 5-3 are written to a static configuration file or to a section within a file.

Table 5-3 Static System Data Parameters

Parameter	Description
HOST_PORT_NUMBER1	The first port number to be used by the Cisco HSI. The default value is 0. **Note** This value must match the peer port setting on the PGW[1] 2200 E-ISUP IPLNK object.
HOST_PORT_NUMBER2	The second port number to be used by the Cisco HSI. The default value is 0. **Note** This value should always be set to 0.
VSCA_IPADDR1	The primary IP address of the primary PGW 2200.
VSCA_IPADDR2	The secondary IP address of the primary PGW 2200. **Note** This value must match that of VSCA_IPADDR1.
VSCB_IPADDR1	The primary IP address of the secondary PGW 2200. **Note** This parameter is not used in a standalone PGW configuration.
VSCB_IPADDR2	The secondary IP address of the secondary PGW 2200. **Note** This value must match that of VSCB_IPADDR1. **Note** This parameter is not used in a standalone PGW configuration.
VSCA_PORT_NUMBER1	The first port number of the primary PGW 2200.
VSCA_PORT_NUMBER2	The second port number of the primary PGW 2200. **Note** This value must match that of VSCA_PORT_NUMBER1.
VSCB_PORT_NUMBER1	The first port number of the secondary PGW 2200. **Note** This parameter is not used in a standalone PGW configuration.
VSCB_PORT_NUMBER2	The second port number of the secondary PGW 2200. **Note** This value must match that of VSCA_PORT_NUMBER2. **Note** This parameter is not used in a standalone PGW configuration.

1. PGW = Public Switched Telephone Network (PSTN) Gateway

Dynamic System Data

To modify the dynamic system data parameters in Table 5-4, use the **sys_config_dynamic** MML name variable for the **prov-add**, **prov-dlt**, and **prov-ed** commands. You need not halt and restart call processing for the changes to take effect.

The parameters in Table 5-4 are written to a dynamic configuration file or to a section within a file.

Table 5-4 Dynamic System Data Parameters

Parameter	Description	Default
LOGDIRECTORY	Defines the directory used when the active log file is created, and also defines the directory where the rotated log file is stored.	/var/log/
LOGFILENAMEPREFIX	Defines the filename prefix used when the log files are created or rotated. The .log postfix is appended to the end of the prefix to establish the name of the active log file.	platform.log
LOGPRIO	This parameter defines the initial logging levels, and by default it is set to TRACE. As soon as the system has initialized and is running, the log levels are determined by the levels set for individual packages (0x0000 to 0xFFFF). See the "Logging Levels" section on page 4-10.	TRACE
LOGFILEROTATESIZE	Triggers a log file rotation based on the size of the active file. The application regularly checks the current size of the file to determine whether a rotation is required. If a file rotation is triggered due to this parameter, the size of the rotated file might be slightly larger than the actual size specified by this parameter. A file rotation triggered by this parameter causes the timer associated with the LOGFILEROTATEINTERVAL parameter to be reset as well.	10 Mb
LOGFILEROTATEINTERVAL	Triggers a log file rotation based on the time elapsed since the previous rotation. This timer is reset after any rotation occurs, regardless of the cause or trigger of the rotation.	1440 minutes (24 hours)
IPADDRNMS	Defines the IP address of the network management system.	—
OVLDSAMPLERATE	Defines the frequency of CPU sampling and threshold checking.	3000 millisecond (ms) polling rate
OVLDLEVEL1PERCENT	Indicates what percentage of calls should be rejected when an overload condition occurs. This parameter is used in conjunction with the OVLDLEVEL1FILTER parameter. The overload level 1 value is the lowest level of overload and must be less than or equal to the provisioned values for OVLDLEVEL2PERCENT and OVLDLEVEL3PERCENT. **Note** If this value is set to zero, no overload level 1 treatment occurs.	20
OVLDLEVEL1FILTER	Indicates what call types should be gapped if an overload level 1 condition occurs. The possible values are: • Normal—Emergency or priority calls are not gapped • All—All calls are gapped, regardless of type **Note** If the overload percentage is set to 100, all calls are gapped irrespective of this setting.	Normal

Table 5-4 Dynamic System Data Parameters (continued)

Parameter	Description	Default
OVLDLEVEL1THRESHLOWER CALLS	Determines the number of active calls below which the application load must fall in order for the overload level 1 condition to be removed.	1800
OVLDLEVEL1THRESHUPPER CALLS	Determines how many simultaneous active calls trigger an overload level 1 condition.	1900
OVLDLEVEL1THRESHLOWER CPU	Determines the CPU utilization level below which the application must fall in order for the overload level 1 condition to be removed.	60
OVLDLEVEL1THRESHUPPER CPU	Determines the level of CPU utilization that triggers an overload level 1 condition.	65
OVLDLEVEL2PERCENT	Indicates what percentage of calls should be rejected when an overload condition occurs. The parameter is used in conjunction with the OVLDLEVEL2FILTER parameter. This is the second level of overload and must be less than or equal to the provisioned value of OVLDLEVEL3PERCENT and greater than or equal to the provisioned value of OVLDLEVEL1PERCENT. **Note** If this value is set to zero, no overload level 1 or 2 treatment occurs (by definition, the level 1 value must also be zero).	75
OVLDLEVEL2FILTER	Indicates what call types should be gapped if an overload level 2 condition occurs (see OVLDLEVEL1FILTER).	Normal
OVLDLEVEL2THRESHLOWER CALLS	Determines the number of active calls below which the application load must fall in order for the overload level 2 condition to be removed.	2000
OVLDLEVEL2THRESHUPPER CALLS	Determines how many simultaneous active calls trigger an overload level 2 condition.	2200
OVLDLEVEL2THRESHLOWER CPU	Determines the level of CPU utilization below which the application must fall in order for the overload level 2 condition to be removed.	70
OVLDLEVEL2THRESHUPPER CPU	Determines the level of CPU utilization that triggers an overload level 2 condition.	80
OVLDLEVEL3PERCENT	Indicates what percentage of calls should be rejected when an overload condition occurs. The parameter is used in conjunction with the OVLDLEVEL3FILTER parameter. This is the highest level of overload and must be greater than or equal to the provisioned values for OVLDLEVEL1PERCENT and OVLDLEVEL2PERCENT. **Note** If this value is set to zero, no overload treatment occurs (by definition, the level 1 and level 2 values must also be zero).	90
OVLDLEVEL3FILTER	Indicates what call types should be gapped if an overload level 3 condition occurs (see OVLDLEVEL1FILTER).	Normal
OVLDLEVEL3THRESHLOWER CALLS	Determines the number of active calls below which the application load must fall in order for the overload level 3 condition to be removed.	2300
OVLDLEVEL3THRESHUPPER CALLS	Determines how many simultaneous active calls trigger an overload level 3 condition.	2400

Table 5-4 Dynamic System Data Parameters (continued)

Parameter	Description	Default
OVLDLEVEL3THRESHLOWER CPU	Determines the level of CPU utilization below which the application must fall in order for the overload level 3 condition to be removed.	85
OVLDLEVEL3THRESHUPPER CPU	Determines the level of CPU utilization that triggers an overload level 3 condition.	95
CIAGENTSCANPERIOD	Defines the frequency with which the CIagent polls the CPU utilization.	—
ALARMDEBOUNCETIME	Defines the length of time that an alarm condition must persist before being reported, and any associated action taken.	0
CALLREFERENCEUSAGE	Determines which call reference identity is passed on to the PGW 2200 (call reference field or Conference ID).	—
DISKUSAGELIMIT	Represents a percentage of disk occupancy. The application continually polls for disk occupancy, and if the percentage rises above the limit set by DISKUSAGELIMIT, the LOW_DISK_SPACE alarm is raised. DISKUSAGELIMIT has a default value of 95 percent. The value range is 0–100, inclusive. When dynamically provisioned, the DISKUSAGELIMIT, if not set within that range, is set to the default value (95) and the CONFIGURATION_ FAILURE alarm is raised.	95

H.323 Stack Configuration

Refer to the *RADVision H.323 Protocol Stack Programmer's Guide* for definitions of each of the RADVision parameters.

The parameter name is based on the ASN.1 paths, but in some cases the parameter name has been shortened for convenience. For example, "capabilities" has been shortened to "caps."

The case of the parameter name reflects exactly the ASN.1 definitions, but is not important to MML configuration.

Nonprovisionable Data

The parameters in Table 5-5 cannot be altered through MML commands.

Table 5-5 Nonprovisionable Data Parameters

H323_SYS	Description
system.manualstart	Present
system.pdlname	Absent
system.delimiter	#FF
ras.gatekeeper	Absent
ras.rasmulticastaddress	224.0.1.41.1718
h245.capabilities.manualoperation	Present

Table 5-5 Nonprovisionable Data Parameters (continued)

H323_SYS	Description
h245.masterslave.manualoperation	Present
q931.manualaccept	Present
q931.earlyH245	Present
q931.autoanswer	Present
q931.manualcallprocessing	Present
q931.h245tunneling	Present

MML Provisionable Data

H.323 System Parameters

The parameters in Table 5-6 are required for H.323 stack initialization. To modify the parameters in Table 5-6, use the **h323_sys** MML name variable for the **prov-add**, **prov-dlt,** and **prov-ed** commands. Halt and restart the application for these changes to take effect.

Note The asterisk (*) after a parameter name in the first column of Table 5-6 denotes a mandatory RADVision parameter that has an inbuilt default value if a value is not set in provisioning.

Table 5-6 H.323 System Initialization Parameters

Parameter	Description	Type	Example
maxCalls*	Maximum number of concurrent calls allowed	INTEGER(0, 65535)	2500
maxChannels*	Maximum number of concurrent channels allowed	INTEGER(0, 65535)	2

Q.931 Parameters

To modify the parameters listed in Table 5-7, use the Q.931 MML name variable for the **prov-add, prov-dlt,** and **prov-ed** commands.

The Update Type column in Table 5-7 shows when the change to a parameter takes effect after it is modified:

- Immediate means that the effect of the change is immediate.
- Start means that the application needs to be restarted for the change to take effect.
- Next Call means that the next call has the new parameter set.

Note Immediate and Next Call update types refer to dynamic system data.

Note The asterisk (*) after a parameter name in the first column of Table 5-7 denotes a mandatory RADVision parameter with an inbuilt default value that will be used if the value is not set in provisioning.

Table 5-7 Q.931 Parameters

Parameter Name	Description	Type	Example	Update Type
responseTimeOut*	The maximum waiting time (in seconds) to receive the first response to a call. If this parameter expires, the call is disconnected.	INTEGER(1,200)	20	Immediate
connectTimeOut*	The maximum time (in seconds) the stack waits for call establishment after the first response is received. If this parameter expires, the call is disconnected.	INTEGER(1,20000)	180	Immediate
callSignalingPort*	The number of the port receiving the calls destined for the PGW 2200.	INTEGER(0,65535)	1720	Start
maxCalls*	The maximum number of simultaneous calls. If this parameter is exceeded, the next call attempt returns busy.	INTEGER(0,65535)	2500	Next Call
notEstablishControl	The stack does not allow the switching of control from the Q.931 to the H.245 stack.	NULL	Not present	Next Call
overlappedSending	Because the Q.931 configuration flag indicates that both parties support overlap sending, this state notifies the other party that it can send an overlap sending message.	NULL	Present	Immediate

Note The Q.931 parameter overlappedSending has been combined with the RAS overlappedSending parameter. The presence of the Q.931 overlappedSending parameter sets the presence of the RAS overlappedSending parameter.

RAS Parameters

The parameters in Table 5-8 are required for RAS stack initialization. To modify the RAS parameters, use the RAS MML name variable for the **prov-add**, **prov-dlt,** and **prov-ed** commands.

The array index [i] in some of the parameter names in the first column of Table 5-8 must be replaced with a valid braced index from 1 to 20, and must be continuous and unique (that is, must contain no duplicates).

The Update Type column in Table 5-8 shows when the change to a parameter takes effect after it is modified:

- Immediate means that the effect of the change is immediate.
- Start means that the application needs to be restarted for the change to take effect.
- Next Call means that the next call has the new parameter set.

Note Immediate and next call update types are dynamic system data.

Note The RAS parameter overlappedSending is not available here because it has been combined with the Q.931 overlappedSending parameter. The presence of the Q.931 overlappedSending parameter sets the presence of the RAS overlappedSending parameter.

Note The asterisk (*) after a parameter name in the first column of Table 5-8 denotes a mandatory RADVision parameter with an inbuilt default value that will be used if the value is not set in provisioning.

Table 5-8 RAS Parameters

Parameter Name	Description	Type	Example	Update Type
manualRAS	If this parameter is present, the stack does not perform automatic RAS procedures (it waits to be driven by the application).	NULL	—	Start
responseTimeOut*	The time (in seconds) that the stack waits until it notifies the application that the called party has failed to respond to a transaction.	INTEGER(1, 200)	10	Immediate
maxFail*	Maximum number of retry gatekeeper registration attempts.	INTEGER(1, 200)	3	Immediate
allowCallsWhenNonReg	If this parameter is present, it allows calls to proceed even if gatekeeper registration has not been done for the PGW 2200.	NULL	Not present	Immediate
manualRegistration	If this parameter is present, the stack does not perform automatic gatekeeper registration procedures (it waits to be driven by the application).	NULL	Not present	Stop/Start
timeToLive	The maximum time (in seconds) the registration of the PGW 2200 with a gatekeeper remains valid. The stack reregisters periodically.	INTEGER(1, 65535)	400	Immediate
rasPort*	Number of the port receiving all RAS transactions for the current endpoint. Set to 0 to allow the OS to look for the available port.	INTEGER(0, 65535)	0	Start

Table 5-8 RAS Parameters (continued)

Parameter Name	Description	Type	Example	Update Type
compare15bitRasCrv	If this parameter is present, it causes the stack to ignore the call reference value (CRV) MSBit in RAS messages.	NULL	—	Immediate
maxRetries*	Maximum number of RAS retransmissions.	INTEGER(1, 200)	3	Immediate
maxMulticastTTL	Maximum number of multicast time to live (TTL).	INTEGER(0, 200)	3	Start
preGrantedArqUse	Choice of direct or routed. If direct, the pregranted admission request (ARQ) feature is used for both direct and routed calls. If routed, the pregranted ARQ feature is used only for routed calls. If absent, the pregranted ARQ is not used.	STRING	direct	Next Call
manualDiscovery.ipAddress	The IP address of a known gatekeeper with which an endpoint might attempt to register.	STRING	10.70.54.53	Start
manualDiscovery.port	The port associated with the manualDiscovery.ipAddress, which can be either a well-known port or another port by agreement.	INTEGER(0, 65535)	1719	Start
gateway.prefix[i]	The telephone prefix for which the gateway is registering as being able to terminate.	STRING	0208	Immediate
gatekeeperId	Identifies the gatekeeper with which the endpoint is trying to register.	STRING	OuterLondon	Immediate
terminalAlias[i].e164	Two variants of the same address; e164 is numeric and h323ID is text by which the endpoint is known.	STRING	0208001000	Immediate
terminalAlias[i].h323ID		STRING	GW@ot.com.au	Immediate
endpointVendor.t35CountryCode	Denote manufacturer of the endpoint.	INTEGER(0, 255)	11	Immediate
endpointVendor.t35Extension		INTEGER(0, 255)	11	Immediate
endpointVendor.manufacturerCode		INTEGER(0, 65535)	9	Immediate

Table 5-8 RAS Parameters (continued)

Parameter Name	Description	Type	Example	Update Type
endpointVendor.productId	Data that the manufacturer assigns to each product.	STRING	H323ESP	Immediate
endpointVendor.versionId	Data that the manufacturer assigns to each version.	STRING	R0.2.4	Immediate

H.245 Parameters

To modify the H.245 parameters listed in Table 5-9, use the H.245 MML name variable for the **prov-add, prov-dlt** and **prov-ed** commands.

The Update Type column in Table 5-9 shows when a change to an H.245 parameter takes effect after it is modified:

- Immediate means that the effect of the change is immediate.
- Start means that the application needs to be restarted for the change to take effect.
- Next Call means that the next call has the new parameter set.

Note Immediate and Next Call update types are dynamic system data.

Table 5-9 H.245 Parameters

Parameter Name	Description	Type	Example	Update Type
masterSlave.terminalType	The terminal type for the PGW 2200.	INTEGER(0, 255)	60	Next Call
masterSlave.manualResponse	If this parameter is present, it cancels automatic acknowledgement of master or slave determination.	NULL	Present	Next Call
masterSlave.timeout	The maximum time (in seconds) the stack waits before it gives up on the master/slave procedure.	INTEGER(0, 65535)	5	Immediate
channelsTimeout	The time (in seconds) the stack waits for a response to a channel establishment message.	INTEGER(0, 65535)	10	Immediate
roundTripTimeout	The time (in seconds) the stack waits for round-trip procedure completion.	INTEGER(0, 65535)	5	Immediate
requestCloseTimeout	The time (in seconds) the stack waits for request close procedure completion.	INTEGER(0, 65535)	5	Immediate

Table 5-9 H.245 Parameters (continued)

Parameter Name	Description	Type	Example	Update Type
requestModeTimeout	The time (in seconds) the stack waits for request mode procedure completion.	INTEGER(0, 65535)	5	Immediate
caps.timeout	The maximum time (in seconds) the stack waits before it gives up on the capability exchange procedure.	INTEGER(0, 65535)	5	Immediate
caps.maxAudioDelay	Maximum H.255 multiplex audio delay jitter.	INTEGER(0, 1023)	60	Immediate
mediaLoopTimeout	The timeout (in seconds) of the media loop procedure.	INTEGER(0, 65535)	5	Immediate

Table 5-10, Table 5-11, and Table 5-12 list the parameters and modes related to configuring codecs. The array index [i] must be replaced with a valid braced index from 1 to 20. The braced index must be continuous and unique (that is, no duplicates).

Table 5-10 H.245 Terminal Capability Codec Parameters

Parameter Name	Type
caps.table[i].entryNo	INTEGER(1, 65535)
caps.table[i].audio.g711Alaw64k	INTEGER(1, 256)
caps.table[i].audio.g711Alaw56k	INTEGER(1, 256)
caps.table[i].audio.g711Ulaw64k	INTEGER(1, 256)
caps.table[i].audio.g711Ulaw56k	INTEGER(1, 256)
caps.table[i].audio.g722at64k	INTEGER(1, 256)
caps.table[i].audio.g722at56k	INTEGER(1, 256)
caps.table[i].audio.g722at48k	INTEGER(1, 256)
caps.table[i].audio.g728	INTEGER(1, 256)
caps.table[i].audio.g729	INTEGER(1, 256)
caps.table[i].audio.g729AnnexA	INTEGER(1, 256)
caps.table[i].audio.g729wAnnexB	INTEGER(1, 256)
caps.table[i].audio.g729AnnexAwAnnexB	INTEGER(1, 256)

Table 5-11 H.245 Channel Codec Parameters

Parameter Name	Type
chan[i].name	STRING
chan[i].audio.g711Alaw64k	INTEGER(1, 256)
chan[i].audio.g711Alaw56k	INTEGER(1, 256)
chan[i].audio.g711Ulaw64k	INTEGER(1, 256)

Table 5-11 H.245 Channel Codec Parameters (continued)

Parameter Name	Type
chan[i].audio.g711Ulaw56k	INTEGER(1, 256)
chan[i].audio.g722at64k	INTEGER(1, 256)
chan[i].audio.g722at56k	INTEGER(1, 256)
chan[i].audio.g722at48k	INTEGER(1, 256)
chan[i].audio.g728	INTEGER(1, 256)
chan[i].audio.g729	INTEGER(1, 256)
chan[i].audio.g729AnnexA	INTEGER(1, 256)
chan[i].audio.g729wAnnexB	INTEGER(1, 256)
chan[i].audio.g729AnnexAwAnnexB	INTEGER(1, 256)

Table 5-12 H.245 Modes

Parameter Name	Type
modes[i].name	STRING
modes[i].audio.g711Alaw64k	NULL
modes[i].audio.g711Alaw56k	NULL
modes[i].audio.g711Ulaw64k	NULL
modes[i].audio.g711Ulaw56k	NULL
modes[i].audio.g722at64k	NULL
modes[i].audio.g722at56k	NULL
modes[i].audio.g722at48k	NULL
modes[i].audio.g728	NULL
modes[i].audio.g729	NULL
modes[i].audio.g729AnnexA	NULL
modes[i].audio.g729wAnnexB	INTEGER(1, 256)
modes[i].audio.g729AnnexAwAnnexB	INTEGER(1, 256)

The following is an example of a configuration file.

Note The configuration file example does not contain a complete list of all configurable items.

```
# $Id: GWmain.base.conf,v 1.7 2001/01/24 06:47:56 arthurv Exp $
#

# This is the base configuration file which will be concatenated together
# with a file derived from questions at install time, to generate the GWmain.conf
# file to be used by the Application GWmain.


##############################################################################
# LOGGING PACKAGE
#
# The Logging package determines the logging level for all defined packages.
# This is a bit mask which controls the 16 debug levels
#
Package = Logging
#
OTLogging = “OFF”#Choice {ON, OFF}. Default: “OFF”

Application  = 0xFF00
CallControl  = 0xFFFF
Connection = 0xFF00
#Connection = 0xFF00
DataManager = 0xFF00
Eisup = 0xFF00
FaultManager = 0xFF00
Gapping = 0xFF00
H323 = 0xFFFF
Infrastructure = 0xFF00
Overload = 0xFF00
ProcessManager = 0xFF00
Provisioning = 0xFE00
Signal = 0xFE00
Snmp  = 0xFE00
SnmpSubagent = 0xFE00
Statistics  = 0xFE00
Trace  = 0xFE00
UserInterface  = 0xFF00



##############################################################################
# H323 SYSTEM Package
#
# Not modifiable at runtime (Static Provisionable Data)
#
Package = H323_SYS
#
maxCalls = 2500
maxChannels = 2



##############################################################################
# Q931 PACKAGE
#
Package = Q931
#
responseTimeOut = 20
```

```
connectTimeOut = 180
callSignalingPort = 1720
maxCalls = 2500
overlappedSending =
earlyH245 =
h245Tunneling =



##############################################################################################
# H323 RAS Package
#
# Modifiable at runtime (Dynamic Provisionable Data) except for manualRAS
#
Package = RAS
#
responseTimeOut = 10
endpointVendor.productID = “GoldWing”
endpointVendor.t35CountryCode = 11
endpointVendor.t35Extension = 11
endpointVendor.manufacturerCode = 9
timeToLive = 400
rasPort = 0
maxRetries = 3
maxMulticastTTL = 3
preGrantedArqUse = direct



##############################################################################################
# H245 PACKAGE
#
# Dynamically Provisionable except for manualOperation(s)
Package = H245
#
channelsTimeout = 10
masterSlave.terminalType = 60
masterSlave.manualOperation =
masterSlave.manualResponse =
masterSlave.timeout = 5
caps.manualOperation =
caps.timeout = 5
caps.maxAudioDelay = 60
caps.table[1].entryNo = 7111
caps.table[1].audio.g711Ulaw64k = 20
caps.table[2].entryNo = 7110
caps.table[2].audio.g711Alaw64k = 20
caps.table[3].entryNo = 728
caps.table[3].audio.g728 = 60
chan[1].name = g711Alaw64k
chan[1].audio.g711Alaw64k = 20
chan[2].name = g711Ulaw64k
chan[2].audio.g711Ulaw64k = 20
chan[3].name = g728
chan[3].audio.g728 = 60
modes[1].name = g711Alaw64k
modes[1].audio.g711Alaw64k =
modes[2].name = g711Ulaw64k
modes[2].audio.g711Ulaw64k =
modes[3].name = g728
modes[3].audio.g728 =
```

```
##########################################################################################
# CALL CONTROL PACKAGE
#
Package = CCPackage
#
Hash = A
Pound = A
Star = B
StopDigit = “#”

A_CC_ChargeInd =  # BCI
A_CC_tEndToEndMethod =
A_CC_tLineUser =
A_CC_tLineStatus =
A_CC_MLC_Action =
A_CC_tSCCPMethod =
A_CC_Interworking =
A_CC_tEndToEndInfAvail =
A_CC_tIsdnAllTheWay =
A_CC_tEchoCancIr =
A_CC_tLineAccess =
A_CC_BNumDataNOA =                # CalledPN
A_CC_BNumDataNPI =
A_CC_BNumDataINN =
A_CC_ANumDataNOA =                # CallingPN
A_CC_Clir =
A_CC_ANumDataSI =
A_CC_ANumDataNPI =
A_CC_A_Cli =
A_CC_oLinecall =  # CallingPC
A_CC_Location =   # CauseInd
A_CC_CodeStandard =
A_CC_ProgressRestrict =           # Event Info
A_CC_oIsdnPref =  # FCI
A_CC_oIsdnAllTheWay =
A_CC_oEndToEndInfAvail =
A_CC_oNatInd =
A_CC_oLSPP =
A_CC_oNBit =
A_CC_oPORC =
A_CC_oPBit =
A_CC_oEndToEndMethod =
A_CC_CollectCallInd =
A_CC_oSCCPMethod =
A_CC_GDES =       # GenericDigits
A_CC_GDTD =
A_CC_NOCI_VC =    # NatureOfConnection
A_CC_NOCI_ECDI =
A_CC_NOCI_CCI =
A_CC_NOCI_SI =
A_CC_TMR =                    # TransmissionMediumRequired
A_CC_INFO_CFN =               # confusion code on INFO receipt
A_CC_GAPPEDCALLCAUSE =  60    # congestion cause for releasing on gapping
A_CC_WAIT_CONFIRM = 30        #20..30 seconds (default is 30), from q764
A_CC_WAIT_ANSWER = 180        #90..180 seconds (default is 180), from q118, refd in

# ----- Cause Codes
------------------------------------------------------------------------------------------
# CC: Call Control, EC: Eisup Cause, HC: H323 Cause

# For the Eisup cause code values see CISCO: EISUP Protocol Specification ENG-46168 version 19
# For the H323 cause code values see ITU-T: Q.850
```

```
# The mappings below are considered constant and not provisionable.
# They can be made provisionable by moving them from the CCPackage
# to the SYS_CONFIG_STATIC package.


# The following is the Eisup to H323 cause code map.
# When the Eisup cause on the left is received from Eisup,
# the H323 cause on right is sent to H323.
# Note: the reverse is not true, this is a one way mapping.
#       The H323 to Eisup cause map is defined further down.

CC_EC_UnallocatedNumber                                          = CC_HC_UnallocatedNumber
CC_EC_NoRouteToTns                                               =
CC_HC_NoRouteToSpecifiedTransitNetwork
CC_EC_NoRouteToDest                          = CC_HC_NoRouteToDestination
CC_EC_SpecialInformationTone                 = CC_HC_SendSpecialInformationTone
CC_EC_MisdialledTkPrefix                     = CC_HC_MisdialedTrunkPrefix
CC_EC_ChUnacceptable                         = CC_HC_ChannelUnacceptable
CC_EC_CallAwardedDeliveredEstCh              = CC_HC_CallAwardedEstablishedChannel
CC_EC_Preemption                             = CC_HC_Preemption
CC_EC_PreemptionCctRes                       = CC_HC_PreemptionCircuitReservedForReuse
CC_EC_NormalClearing                         = CC_HC_NormalCallClearing
CC_EC_UserBusy                               = CC_HC_UserBusy
CC_EC_NoUserResponding                       = CC_HC_NoUserresponding
CC_EC_NoAnswerAlertedUser                    = CC_HC_NoAnswerFromAlertedUser
CC_EC_SubAbscent                             = CC_HC_SubscriberAbsent
CC_EC_CallRejected                           = CC_HC_CallRejected
CC_EC_NumberChanged                          = CC_HC_NumberChanged
CC_EC_RedirectionToNewDest                   = CC_HC_RedirectionToNewDestination
CC_EC_RoutingError                           = CC_HC_ExchangeRouteError
CC_EC_NonSelectedUserClearing                = CC_HC_NonSelectedUserClearing
CC_EC_DestOutOfOrder                         = CC_HC_DestinationOutOfOrder
CC_EC_InvalidNumberFormat                    = CC_HC_InvalidNumberFormat
CC_EC_FacilityRejected                       = CC_HC_FacilityRejected
CC_EC_ResponseToStatusEnquiry                = CC_HC_ResponceToStatusEnquiry
CC_EC_NormalUnspecified                      = CC_HC_NormalUnspecified
CC_EC_NoCircuitAvailable                     = CC_HC_NoCircuitChannelAvailable
CC_EC_NetworkOutOfOrder                      = CC_HC_NetworkOutOfOrder
CC_EC_PermanentFrameModeOos                  = CC_HC_PermanentFrameModeConnectionOutOfService
CC_EC_PermanentFrameModeOperational          = CC_HC_PermanentFrameModeConnectionOperational
CC_EC_TemporaryFailure                       = CC_HC_TemporaryFailure
CC_EC_SwitchingEquipCongestion               = CC_HC_SwitchingEquipmentCongestion
CC_EC_AccessInfoDiscarded                    = CC_HC_AccessInformationDiscarded
CC_EC_ReqCircuitUnavail                                          =
CC_HC_RequestedCircuitChannelNotAvailable
CC_EC_PrecedenceBlocked                      = CC_HC_PrecedenceCallBlocked
CC_EC_ResourcesUnavailUnspec                 = CC_HC_ResourceUnavailable
CC_EC_QualityUnavail                         = CC_HC_QualityOfServiceNotAvailable
CC_EC_ReqFacilityNotSubscr                   = CC_HC_RequestedFacilityNotSubscribed
CC_EC_OutgoingCallsBarredInCug               = CC_HC_OutgoingCallsBaredWithinCUG
CC_EC_IncomingCallsBarredInCug               = CC_HC_IncomingCallsBaredWithinCUG
CC_EC_BearcapNotAuthorized                   = CC_HC_BearerCapabilityNotAuthorized
CC_EC_BaercapNotAvail                        = CC_HC_BearerCapabilityNotPresentlyAvailable
CC_EC_InconOutgoingAccAndSubClass            = CC_HC_InconsistencyAcessInfoSubscriberClass
CC_EC_ServiceOrOptionNotAvail                = CC_HC_ServiceOrOptionUnavailable
CC_EC_BearcapNotImp                          = CC_HC_BearerCapabilityNotImplemented
CC_EC_ChTypeNotImp                           = CC_HC_ChannelTypeNotImplemented
CC_EC_ReqFacilityNotImp                      = CC_HC_RequestedFacilityNotImplemented
CC_EC_OnlyRestrictDigInfoBearer              = CC_HC_OnlyRestrictedDigitalBearerInfoCapability
CC_EC_ServiceOrOptionNotImpUnspec            = CC_HC_ServiceOrOptionNotImplemented
CC_EC_InvalidCallReferenceValue              = CC_HC_InvalidCallreferenceValue
CC_EC_ChIdNotExist                           = CC_HC_IdentifiedChannelDoesnotExist
CC_EC_SuspendExistButNotThisId               = CC_HC_ASuspendedCallExistsThisCallIdDoesNot
CC_EC_CallIdInUse                            = CC_HC_CallIdentityInUse
```

```
CC_EC_NoCallSuspended                       = CC_HC_NoCallSuspended
CC_EC_CallIdHasBeenCleared                  = CC_HC_CallHavingTheRequestedCallIdHasBeenCleared
CC_EC_UserNotMemberOfCug                    = CC_HC_UserNotMemberOfCUG
CC_EC_IncompatibleDest                      = CC_HC_IncompatibleDestination
CC_EC_NonExistentCug                        = CC_HC_NonExistantCUG
CC_EC_InvalidTns                            = CC_HC_InvalidTransitNetworkSelection
CC_EC_InvalidMsgUnspec                      = CC_HC_InvalidMessage
CC_EC_MandatoryElementMissing               = CC_HC_MandatoryInformationElementIsMissing
CC_EC_MsgTypeNotImp                         = CC_HC_MessageTypeNonExistantOrNotImplemented
CC_EC_MsgTypeNotImpOrWrongState             = CC_HC_MessageTypeNotCompatibleWithState
                                              OrNonExistantOrNotImplemented
CC_EC_ElemTypeNotImp                        = CC_HC_InformationElementParameterNonExistantOrNotImplemented
CC_EC_InvalidElemContents                   = CC_HC_InvalidInformationElementContents
CC_EC_MsgInWrongState                       = CC_HC_MessageNotCompatibleWithCallState
CC_EC_RecoveryOnTimerExpiry                 = CC_HC_RecoveryOnTimerExpiry
CC_EC_ParamUnrecPassed                      = CC_HC_ParameterNonExistantOrNotImplementedPassedOn


# When the H323 cause on the left is received from H323,
# the Eisup cause on the right is sent to Eisup.
# Note: the reverse is not true, this is a one way mapping.
# The Eisup to H323 cause map is defined in above.

CC_HC_UnallocatedNumber                                     = CC_EC_UnallocatedNumber
CC_HC_NoRouteToSpecifiedTransitNetwork                       = CC_EC_NoRouteToTns
CC_HC_NoRouteToDestination                                   = CC_EC_NoRouteToDest
CC_HC_SendSpecialInformationTone                            = CC_EC_SpecialInformationTone
CC_HC_MisdialedTrunkPrefix                                   = CC_EC_MisdialledTkPrefix
CC_HC_ChannelUnacceptable                                    = CC_EC_ChUnacceptable
CC_HC_CallAwardedEstablishedChannel                          = CC_EC_CallAwardedDeliveredEstCh
CC_HC_Preemption                                             = CC_EC_Preemption
CC_HC_PreemptionCircuitReservedForReuse                      = CC_EC_PreemptionCctRes
CC_HC_NormalCallClearing                                     = CC_EC_NormalClearing
CC_HC_UserBusy                                               = CC_EC_UserBusy
CC_HC_NoUserresponding                                       = CC_EC_NoUserResponding
CC_HC_NoAnswerFromAlertedUser                                = CC_EC_NoAnswerAlertedUser
CC_HC_SubscriberAbsent                                       = CC_EC_SubAbscent
CC_HC_CallRejected                                           = CC_EC_CallRejected
CC_HC_NumberChanged                                          = CC_EC_NumberChanged
CC_HC_RedirectionToNewDestination                            = CC_EC_RedirectionToNewDest
CC_HC_ExchangeRouteError                                     = CC_EC_RoutingError
CC_HC_NonSelectedUserClearing                                = CC_EC_NonSelectedUserClearing
CC_HC_DestinationOutOfOrder                                  = CC_EC_DestOutOfOrder
CC_HC_InvalidNumberFormat                                    = CC_EC_InvalidNumberFormat
CC_HC_FacilityRejected                                       = CC_EC_FacilityRejected
CC_HC_ResponceToStatusEnquiry                                = CC_EC_ResponseToStatusEnquiry
CC_HC_NormalUnspecified                                      = CC_EC_NormalUnspecified
CC_HC_NoCircuitChannelAvailable                              = CC_EC_NoCircuitAvailable
CC_HC_NetworkOutOfOrder                                      = CC_EC_NetworkOutOfOrder
CC_HC_PermanentFrameModeConnectionOutOfService               = CC_EC_PermanentFrameModeOos
CC_HC_PermanentFrameModeConnectionOperational                = CC_EC_PermanentFrameModeOperational
CC_HC_TemporaryFailure                                       = CC_EC_TemporaryFailure
CC_HC_SwitchingEquipmentCongestion                           = CC_EC_SwitchingEquipCongestion
CC_HC_AccessInformationDiscarded                             = CC_EC_AccessInfoDiscarded
CC_HC_RequestedCircuitChannelNotAvailable                    = CC_EC_ReqCircuitUnavail
CC_HC_PrecedenceCallBlocked                                  = CC_EC_PrecedenceBlocked
CC_HC_ResourceUnavailable                                    = CC_EC_ResourcesUnavailUnspec
CC_HC_QualityOfServiceNotAvailable                           = CC_EC_QualityUnavail
```

```
CC_HC_RequestedFacilityNotSubscribed                                    = CC_EC_ReqFacilityNotSubscr
CC_HC_OutgoingCallsBaredWithinCUG                                       = CC_EC_OutgoingCallsBarredInCug
CC_HC_IncomingCallsBaredWithinCUG                                       = CC_EC_IncomingCallsBarredInCug
CC_HC_BearerCapabilityNotAuthorized                                     = CC_EC_BearcapNotAuthorized
CC_HC_BearerCapabilityNotPresentlyAvailable                             = CC_EC_BaercapNotAvail
CC_HC_InconsistencyAcessInfoSubscriberClass                             = CC_EC_InconOutgoingAccAndSubClass
CC_HC_ServiceOrOptionUnavailable                                        = CC_EC_ServiceOrOptionNotAvail
CC_HC_BearerCapabilityNotImplemented                                    = CC_EC_BearcapNotImp
CC_HC_ChannelTypeNotImplemented                                         = CC_EC_ChTypeNotImp
CC_HC_RequestedFacilityNotImplemented                                   = CC_EC_ReqFacilityNotImp
CC_HC_OnlyRestrictedDigitalBearerInfoCapability                         = CC_EC_OnlyRestrictDigInfoBearer
CC_HC_ServiceOrOptionNotImplemented                                     = CC_EC_ServiceOrOptionNotImpUnspec
CC_HC_InvalidCallreferenceValue                                         = CC_EC_InvalidCallReferenceValue
CC_HC_IdentifiedChannelDoesnotExist                                     = CC_EC_ChIdNotExist
CC_HC_ASuspendedCallExistsThisCallIdDoesNot                             = CC_EC_SuspendExistButNotThisId
CC_HC_CallIdentityInUse                                                 = CC_EC_CallIdInUse
CC_HC_NoCallSuspended                                                   = CC_EC_NoCallSuspended
CC_HC_CallHavingTheRequestedCallIdHasBeenCleared                        = CC_EC_CallIdHasBeenCleared
CC_HC_UserNotMemberOfCUG                                                = CC_EC_UserNotMemberOfCug
CC_HC_IncompatibleDestination                                           = CC_EC_IncompatibleDest
CC_HC_NonExistantCUG                                                    = CC_EC_NonExistentCug
CC_HC_InvalidTransitNetworkSelection                                    = CC_EC_InvalidTns
CC_HC_InvalidMessage                                                    = CC_EC_InvalidMsgUnspec
CC_HC_MandatoryInformationElementIsMissing                              = CC_EC_MandatoryElementMissing
CC_HC_MessageTypeNonExistantOrNotImplemented                            = CC_EC_MsgTypeNotImp
CC_HC_MessageTypeNotCompatibleWithStateOrNonExistantOrNotImplemented    = CC_EC_MsgTypeNotImpOrWrongState
CC_HC_InformationElementParameterNonExistantOrNotImplemented = CC_EC_ElemTypeNotImp
CC_HC_InvalidInformationElementContents                                 = CC_EC_InvalidElemContents
CC_HC_MessageNotCompatibleWithCallState                                 = CC_EC_MsgInWrongState
CC_HC_RecoveryOnTimerExpiry                                             = CC_EC_RecoveryOnTimerExpiry
CC_HC_ParameterNonExistantOrNotImplementedPassedOn                      = CC_EC_ParamUnrecPassed

#
###############################################################################################
# FAULTMANAGEMENT PACKAGE
#
Package = FaultManagement
#
FMRaiseRecoveryAction  =   “ON”
FMClearRecoveryAction  =   “ON”

###############################################################################################
# GAPPING PACKAGE
#
# Set the gapping percentage level for each side. A level of 0 indicates no gapping
# A level of 100 indicates gap all calls (except priority calls - see treatment below)
#
Package = Gapping
#
H323level = 0
EISUPlevel = 0

#
# Priority treatment determines the treatment of priority calls during gapping.
# GapAlways indicates priority calls are treated as normal calls
# GapNever indicate priority calls are never to be gapped
# GapOn100PercentGapping indicates priority calls are only gapped when 100 percent
# gapping is applied.
#

#PriorityCallTreatment = GapOn100PercentGapping
#PriorityCallTreatment = GapNever
PriorityCallTreatment = GapAlways
```

```
##############################################################################################
# APPLICATION PACKAGE
#
#
Package = Application
#
DefaultCallProcessingStatus = “UP”  #Choice {“UP”, “DOWN”}

WaitBeforeCallReleaseTimer  = 60    #Default is 60
RestartPendingTimer         = 60    #Default is 60
HaltPendingTimer            = 60    #Default is 60
RebootPendingTimer          = 60    #Default is 60



##############################################################################################
# DYNAMIC SYSTEM DATA
#
#
Package = SYS_CONFIG_DYNAMIC

# Logging
#
LOGDIRECTORY          = “var/log/”       #Default: “var/log/”
LOGFILENAMEPREFIX     = “platform”       #Default: “platform.log”
LOGPRIO                   = “TRACE”          #Choice {DEBUG, TRACE, INFO, WARN, ERR, CRIT}. Default: “WARN”
LOGFILEROTATESIZE     = 10240            #Default: 10240 bytes (10Mb)
LOGFILEROTATEINTERVAL = 1440             #Default: 1440 min (24hrs)


# Overload
#
OVLDSAMPLERATE             = 3000        #Default: 3000 msec polling rate

OVLDLEVEL1PERCENT          = 20          #Default: 99
OVLDLEVEL1FILTER           = “NORMAL”    #Choice {“NORMAL”, “ALL”}. Default: “NORMAL”
OVLDLEVEL1THRESHUPPERCPU   = 65          #Default: 100
OVLDLEVEL1THRESHLOWERCPU   = 60          #Default: 100
OVLDLEVEL1THRESHUPPERCALLS = 1900        #Default: 1000
OVLDLEVEL1THRESHLOWERCALLS = 1800        #Default: 1000

OVLDLEVEL2PERCENT          = 75          #Default: 99
OVLDLEVEL2FILTER           = “NORMAL”    #Choice {“NORMAL”, “ALL”}. Default: “NORMAL”
OVLDLEVEL2THRESHUPPERCPU   = 80          #Default: 100
OVLDLEVEL2THRESHLOWERCPU   = 70          #Default: 100
OVLDLEVEL2THRESHUPPERCALLS = 2200        #Default: 1000
OVLDLEVEL2THRESHLOWERCALLS = 2000        #Default: 1000

OVLDLEVEL3PERCENT          = 90          #Default: 99
OVLDLEVEL3FILTER           = “NORMAL”    #Choice {“NORMAL”, “ALL”}. Default: “NORMAL”
OVLDLEVEL3THRESHUPPERCPU   = 95          #Default: 100
OVLDLEVEL3THRESHLOWERCPU   = 85          #Default: 100
OVLDLEVEL3THRESHUPPERCALLS = 2400        #Default: 1000
OVLDLEVEL3THRESHLOWERCALLS = 2300        #Default: 1000
DISKUSAGELIMIT             = 98          #Default: 95% Disk Usage
```

```
##############################################################################################
#
Package = SYS_CONFIG_STATIC


# Call Control

# For the Eisup cause code values see CISCO: EISUP Protocol Specification ENG-46168 version 19
# For the H323 cause code values see ITU-T: Q.850

# The default cause codes, used when there is no map entry for a received cause

CC_EC_DEFAULT                              = CC_EC_NormalUnspecified
CC_HC_DEFAULT                              = CC_HC_NormalUnspecified

# Unassigned Eisup cause codes
#

CC_EC_AccessBarred                       = CC_HC_DEFAULT
CC_EC_Acknowledgement                    = CC_HC_DEFAULT
CC_EC_AddressIncomplete                  = CC_HC_DEFAULT
CC_EC_AnonymousCallRejection             = CC_HC_DEFAULT
CC_EC_BlacklistBNumberMatched            = CC_HC_DEFAULT
CC_EC_BlacklistCliLengthInvalid          = CC_HC_DEFAULT
CC_EC_BlacklistCliMatched                = CC_HC_DEFAULT
CC_EC_BlacklistCpcRestricted             = CC_HC_DEFAULT
CC_EC_BlacklistNoCli                     = CC_HC_DEFAULT
CC_EC_BlacklistNoaRestricted             = CC_HC_DEFAULT
CC_EC_Busy                               = CC_HC_DEFAULT
CC_EC_CallRejectCallGapping              = CC_HC_DEFAULT
CC_EC_CallTerminated                     = CC_HC_DEFAULT
CC_EC_CallTypeIncompatible               = CC_HC_DEFAULT
CC_EC_CallingDroppedWhileOnHold          = CC_HC_DEFAULT
CC_EC_CallingPartyOffHold                = CC_HC_DEFAULT
CC_EC_ChannelOutOfService                = CC_HC_DEFAULT
CC_EC_Congestion                         = CC_HC_DEFAULT
CC_EC_CotFailure                         = CC_HC_DEFAULT
CC_EC_CugAccessBarred                    = CC_HC_DEFAULT
CC_EC_DteControlledNotReady              = CC_HC_DEFAULT
CC_EC_DteUncontrolledNotReady            = CC_HC_DEFAULT
CC_EC_ExcessiveDigCallProceeding         = CC_HC_DEFAULT
CC_EC_FacilityNotRegistered              = CC_HC_DEFAULT
CC_EC_FlowControlledCongestion           = CC_HC_DEFAULT
CC_EC_GroupRestrictions                  = CC_HC_DEFAULT
CC_EC_IncomingCallsBarred                = CC_HC_DEFAULT
CC_EC_InterceptedSubscriber              = CC_HC_DEFAULT
CC_EC_InterworkUnspec                    = CC_HC_DEFAULT
CC_EC_InvalidCallRef                     = CC_HC_DEFAULT
CC_EC_MesgWithUnrecElemDiscarded         = CC_HC_DEFAULT
CC_EC_MessageNotUnderstood               = CC_HC_DEFAULT
CC_EC_MisroutedCallPortedNumber          = CC_HC_DEFAULT
CC_EC_NetworkAddressExtensionError       = CC_HC_DEFAULT
CC_EC_NetworkTermination                 = CC_HC_DEFAULT
CC_EC_NewDestination                     = CC_HC_DEFAULT
CC_EC_NumberUnobtainable                 = CC_HC_DEFAULT
CC_EC_OperatorPriorityAccess             = CC_HC_DEFAULT
CC_EC_OutOfCatchmentArea                 = CC_HC_DEFAULT
CC_EC_OutgoingCallsBarred                = CC_HC_DEFAULT
CC_EC_PermanentIcb                       = CC_HC_DEFAULT
CC_EC_PortedNumber                       = CC_HC_DEFAULT
CC_EC_PreemptionCctUnavailable           = CC_HC_DEFAULT
CC_EC_Prefix0DialledInError              = CC_HC_DEFAULT
CC_EC_Prefix1DialledInError              = CC_HC_DEFAULT
CC_EC_Prefix1NotDialled                  = CC_HC_DEFAULT
```

```
CC_EC_PriorityForcedRelease              = CC_HC_DEFAULT
CC_EC_Proprietary                        = CC_HC_DEFAULT
CC_EC_ProtErrThresholdExceeded           = CC_HC_DEFAULT
CC_EC_ProtocolErrorUnspec                = CC_HC_DEFAULT
CC_EC_Reject                             = CC_HC_DEFAULT
CC_EC_RejectedDivertedCall               = CC_HC_DEFAULT
CC_EC_RemoteProcError                    = CC_HC_DEFAULT
CC_EC_RepeatAttempt                      = CC_HC_DEFAULT
CC_EC_RouteOutOfService                  = CC_HC_DEFAULT
CC_EC_SelectiveCallBarring               = CC_HC_DEFAULT
CC_EC_ServiceIncompatible                = CC_HC_DEFAULT
CC_EC_ServiceTemporarilyUnavailable      = CC_HC_DEFAULT
CC_EC_ServiceUnavailable                 = CC_HC_DEFAULT
CC_EC_SignalNotUnderstood                = CC_HC_DEFAULT
CC_EC_SignalNotValid                     = CC_HC_DEFAULT
CC_EC_SignalingSystemIncompatible       = CC_HC_DEFAULT
CC_EC_SubControlledIcb                    = CC_HC_DEFAULT
CC_EC_SubNotFoundDle                     = CC_HC_DEFAULT
CC_EC_SubscriberCallTerminate            = CC_HC_DEFAULT
CC_EC_SubscriberIncompatible             = CC_HC_DEFAULT
CC_EC_SubscriberMoved                    = CC_HC_DEFAULT
CC_EC_SubscriberOutOfService             = CC_HC_DEFAULT
CC_EC_TemporaryOos                       = CC_HC_DEFAULT
CC_EC_TerminalCongestion                 = CC_HC_DEFAULT
CC_EC_Transferred                        = CC_HC_DEFAULT
CC_EC_TranslationOos                     = CC_HC_DEFAULT
CC_EC_UnallocatedDestNumber              = CC_HC_DEFAULT
CC_EC_UndefinedBg                        = CC_HC_DEFAULT
CC_EC_Unknown                            = CC_HC_DEFAULT
CC_EC_UnrecElemPassedOn                  = CC_HC_DEFAULT
CC_EC_VacentCode                         = CC_HC_DEFAULT
CC_EC_WhitelistCliNotMatched             = CC_HC_DEFAULT

########################################
# INSTALLATION DERIVED CONFIG PARAMETERS
Package = RAS
manualDiscovery.ipAddress = “10.70.54.53”
manualDiscovery.port = “1719”
gateway.prefix[1] = “0208”
terminalAlias[1].h323ID = “cisco@OuterLondonDomain.com”
gatekeeperId = “OuterLondon”
endpointVendor.versionID = “0.2.3”
########################################
Package = SYS_CONFIG_STATIC
HOST_PORT_NUMBER1= “8003”
VSC1_NAME= “goliath”
VSC1_PORT_NUMBER=”8003”
NodeId = “H323-GW1”
########################################
Package = Application
Hardware = “Sun Netra T1”
Location = “H323 - GW1”
########################################
```