



## Signaling System No. 7 Messages

---

Signaling System No. 7 (SS7) messages describe error and status messages associated with American National Standards Institute/International Telecommunication Union (ANSI/ITU) and Integrated Services Digital Network User Part (ISUP) applications.

SS7 messages appear in the log directory (\$XNV/log) in a file named in the following manner: cktint-Mmmdd.log. The mnemonic Mmm represents the month and dd represents the day of the month. For example, the log filename for April 23 is designated as cktint-Apr23.log. SS7 messages are described in the following manner:

nnn: Message

**Explanation** An explanation of the message.

**Action** A description of the action the user should take.

### SS7 Messages

001: Couldn't open the Parameter Order Configuration File

**Explanation** The cktint was not able to open the parameter order configuration file.

**Action** Make sure that param\_ord.cfg file is present in the \$XNV directory.

012: TX to VCO failed, no link connected

**Explanation** The internally generated message could not be transmitted to the SDS/VCO since no link is connected.

**Action** Verify that the VCO link is configured properly in the CktInt.cfg configuration file in the \$XNV directory. Run the script tcp-links to verify the status of the host and VCO links. If the host link is offline, connect a host.

015: HOST TX to VCO failed

**Explanation** The transmission of a host message to the SDS/VCO through cktint has failed while writing to the message queue.

**Action** Verify that all processes are running. If they are all running, verify that the total number of messages outstanding in all the message queues together have not exceeded the system configuration limit (1600) by running the command `ipcs -ob` from the command line. If they have not exceeded the limit, verify that the host links are connected. If they are connected, bring down cktint and the stack and bring them up again.

016: TX to VCO failed

**Explanation** The internally generated cktint message has failed to reach the SDS/VCO due to a queue write error.

**Action** Verify that all processes are running. If they are all running, verify that the total number of messages outstanding in all the message queues together have not exceeded the system configuration limit (1600) by running the command `ipcs -ob` from the command line. If they have not exceeded the limit, verify that the host links are connected. If they are connected, bring down cktint and the stack and bring them up again.

017: TX to HOST failed

**Explanation** The cktint report to host has failed due to queue write error.

**Action** Verify that all processes are running. If they are all running, verify that the total number of messages outstanding in all the message queues together have not exceeded the system configuration limit (1600) by running the command `ipcs -ob` from the command line. If they have not exceeded the limit, verify that the host links are connected. If they are connected, bring down cktint and the stack and bring them up again.

023: ckt\_ss7\_to\_sds - Invalid group number

**Explanation** An invalid group number is detected in the circuit configuration data.

**Action** Make sure that the trunk group number specified in the `ckt_ss7_to_sds` file is also configured in the `grp_ss7_to_sds` file.

026: Invalid SS7 Primitive Detected

**Explanation** An invalid SS7 primitive has been received by the cktint from the EBS (ADC NewNet) stack.

**Action** Verify the primitive number and report it to Cisco Systems TAC.

## 027: Undefined CIC Received

**Explanation** An undefined circuit identification code (CIC) has been received from the EBS (ADC NewNet) stack.

**Action** Ensure that all circuits configured in the EBS (ADC NewNet) stack are also configured in `cktint`.

## 028: Invalid Circuit Maintenance event detected

**Explanation** An invalid SS7 circuit maintenance message is received from the network.

**Action** Verify the message type and ensure that it is a valid ISUP message.

## 029: Circuit Validation Test Failed

**Explanation** The circuit validation response for an outgoing circuit validation test indicates failure.

**Action** Verify this failure notification with the network operator.

## 031: Circuit Query State Mismatch

**Explanation** There is an inconsistency between the local state and the remote state of a circuit.

**Action** Reset the circuit through `isup_console`.

## 032: Circuit Query Response, Undefined CIC

**Explanation** Circuit query response message with undefined circuit identification code (CIC) has been received from the EBS (ADC NewNet) stack.

**Action** Verify the circuit configuration on both `cktint` and the EBS (ADC NewNet) stack.

## 033: Circuit Reset, Undefined CIC

**Explanation** Circuit reset message has been received from the EBS (ADC NewNet) stack on a circuit which is not configured in `cktint`.

**Action** Verify the circuit configuration on both `cktint` and the EBS (ADC NewNet) stack to ensure there is no inconsistency.

## 035: Maximum circuits exceeded in CKT GRP

**Explanation** The maximum supported circuits per trunk group has been exceeded.

**Action** In the `ckt_ss7_to_sds` configuration file, check that each trunk group has only a maximum of 24 circuits for ANSI with circuit ID 0 to 23 and a maximum of 32 circuits for ITU with circuit ID 0 to 31.

037: Interworking TX to PORTICO failed

**Explanation** A cktint-generated SS7 message to the EBS (ADC NewNet) stack failed while writing to the queue.

**Action** Try to reduce the length of the SS7 Network Message Generation (\$49) command from the host for that particular message if the number of parameters is greater than 14, for releases before CCITT V5.1 FSR02 for ITU. If the problem persists, verify that all processes are running. If they are all running, verify that the total number of messages outstanding in all the message queues together have not exceeded the system configuration limit (1600) by running the command `ipcs -ob` from the command line. If they have not exceeded the limit, verify that the host links are connected. If they are connected, bring down cktint and the stack and bring them up again.

039: hunt\_circuit( ) - Hunt Failed

**Explanation** The hunt failed to find an available circuit in the resource group specified. This error is seen when there is no resource available in that resource group.

**Action** The SS7 Network Message Generation (\$49) command will be rejected with a network status byte (NSB) of 0xC9 (NSB\_NO\_RESOURCE). The host application should handle this NSB.

040: load\_params()- Undefined Parameter Name Found:

**Explanation** An undefined parameter has been received from the network.

**Action** None required.

042: Resource Group Size Exceeded, Group number:

**Explanation** The number of ports in the resource group has exceeded the maximum number of ports allowed in a resource group.

**Action** Verify that the `res_grp.cfg` file resided in the \$XNV directory and ensure that the number of ports in the resource group number printed in the error message has not exceeded the maximum limit. Refer to the appropriate SS7 supplement for information on maximum number of ports that can be configured in each resource group.

043: build\_resource\_group, System:

**Explanation** An error was encountered when building the resource groups. The resource group number has exceeded the maximum limit.

**Action** Verify that the `res_grp.cfg` file resides in the \$XNV directory and ensure that the resource group number has not exceeded the maximum number of allowed resource groups. Refer to the appropriate SS7 supplement for the resource group range.

044: Undefined SS7 Message Requested

**Explanation** An undefined SS7 message has been received from the host.

**Action** Verify the message type and make sure the host application does not send this message. If it is a valid message, report this situation to the Cisco Systems TAC.

045: SS7 Protocol Violation Msg

**Explanation** An SS7 Network Message Generation (\$49) command with an unexpected ISUP message was received from the host.

**Action** Verify the SS7 protocol specification to determine whether this message is allowed in that particular call state. If it is allowed, report this situation to the Cisco Systems TAC. Otherwise, ensure that the application does not send this message during that call state.

046: Invalid Template Specified:

**Explanation** This error is seen in two scenarios:

- a. An invalid SS7 message template has been specified in the SS7 Network Message Generation (\$49) command.
- b. An invalid SS7 message template number has been specified in the CktInt.cfg file.

**Action** In the first scenario (a.) ensure that the required template file specified exists in the \$XNV/templates directory. In the second scenario (b.) ensure that the template number specified the cktint.cfg file is a valid template number.

048: Non existent template number specified

**Explanation** A nonexistent template number is specified in the default\_templates file.

**Action** Verify that a template file exists for the template number specified in the default\_templates file.

049: unload\_params( ) - Undefined Parameter Name Found

**Explanation** An undefined parameter has been found in an ISUP message from the host.

**Action** If the host sends this undefined parameter to the network, ensure that this parameter is added in the param-ord.cfg file.

051: unload\_params - Parameter Count Too Large

**Explanation** The parameter count in the SS7 Network Message Generation (\$49) command's ISUP message is greater than the maximum supported value of 64.

**Action** Reduce the parameters sent in the \$49 command.

052: unload\_params - Parameter Size Too Large

**Explanation** The parameter length in the SS7 Network Message Generation (\$49) command's ISUP message is greater than the maximum supported value of 255. The \$49 command will be rejected with a network status byte of 0xCB (NSB\_PARAM\_ERROR\_IN\_SEGMENT).

**Action** Verify the length of the parameter in the \$49 command.

055: Process( ) - Outgoing Continuity Test Failed On CIC:

**Explanation** The outgoing continuity test has failed on the circuit identification code (CIC).

**Action** Verify the voice path.

056: ISUP Parameter from Host Exceeds Max Length (Using Max)

**Explanation** The cktint process has received an ISUP message from the host which contains a parameter whose length is greater than the maximum supported length for that particular parameter. The SS7 Network Message Generation (\$49) command will be rejected with a network status byte of 0xCB (NSB\_PARAM\_ERROR\_IN\_SEGMENT).

**Action** Verify and correct the length of the parameter in the \$49 command.

057: ISUP Parameter from Network Exceeds Max Length (Using Max):

**Explanation** The cktint process has received an ISUP message from the network which contains a parameter whose length is greater than the maximum supported length for that particular parameter. In this case, cktint strips the length to the maximum supported parameter length.

**Action** Verify with the network provider the sending of a parameter with an invalid length.

061: Couldn't open the default 'default\_templates' File

**Explanation** The cktint could not open the default templates file.

**Action** Verify that the default\_templates file exists in \$XNV/templates directory.

063: Couldn't open the Group Mapping Configuration File

**Explanation** The cktint could not open the group mapping configuration file grp\_ss7\_ss7\_to\_sds.

**Action** Verify that a proper grp\_ss7\_to\_sds file exists in the \$XNV directory.

064: Couldn't open the Circuit Mapping Configuration File

**Explanation** The cktint could not open the circuit mapping configuration file ckt\_ss7\_to\_sds.

**Action** Verify that a proper ckt\_ss7\_to\_sds file exists in the \$XNV directory.

065: grp\_ss7\_to\_sds - Out of Range Group Number

**Explanation** The group number is out of range in the group configuration file.

**Action** Correct the trunk group number to a proper value. Refer to the appropriate SS7 supplement for the range of valid trunk group numbers.

066: ckt\_ss7\_to\_sds - Invalid Circuit

**Explanation** The port number specified in the ckt\_ss7\_to\_sds file is invalid.

**Action** Configure a valid VCO/4K port address. Refer to the appropriate SS7 supplement for configuration information.

069: ProcArgs( ) - Invalid DEBUG switch value

**Explanation** The debug switch value specified in the isup\_console 'd' option is invalid.

**Action** Specify a valid debug number. Refer to the appropriate SS7 supplement for instructions.

070: Ckt( ) - Unable to GET the CKTINT environment variable...

**Explanation** The cktint failed to get the XNV environment variable.

**Action** Log out and then log in as cktint. Run start-ss7.sh.

071: Ckt( ) - Failed trying to change directories to the CKTINT execution directory..."

**Explanation** The cktint failed to change to the \$XNV directory.

**Action** Verify that the \$XNV directory exists. If does not exist, reinstall the software.

073: ckt\_config()- Empty Circuit Mapping Configuration File:

**Explanation** The ckt\_ss7\_to\_sds file is empty.

**Action** Configure valid circuits in the ckt\_ss7\_to\_sds file.

079: ChildReg( ) - SS7 TX process FAILED...

**Explanation** The SS7 transmit process \_ssisan has exited after failing to register with EBS (ADC NewNet) stack.

**Action** Verify that the EBS (ADC NewNet) stack is running. If it is running, bring down cktint and the EBS stack and restart them.

080: ChildReg( ) - SS7 RCV process FAILED...

**Explanation** The SS7 receive process `_ssisan` has exited after failing to register with the EBS (ADC NewNet) stack.

**Action** Verify that the EBS stack is running. If it is running, bring down `cktint` and the EBS stack and restart them.

084: VCO command received. Cktint initiated - REJECTED/FAILED.

**Explanation** The `cktint` generated VCO commands (\$70, \$66, etc.) were rejected by the VCO.

**Action** Identify the network status byte (NSB) value in the rejected command and analyze the VCO configuration to determine the reason that this command was rejected. Verify that the T1 hardware status is in service.

090: Host generated CGB is missing Range and Status parameter

**Explanation** The circuit group blocking (CGB) message generated by the host does not contain the mandatory range and status parameter.

**Action** Make sure the SS7 Network Message Generation (\$49) command from the host for the CGB message contains the range and status parameter.

091: Host generated CGU is missing Range and Status parameter

**Explanation** The host-generated circuit group unblocking (CGU) ISUP message does not contain the mandatory range and status parameter.

**Action** Verify that the SS7 Network Message Generation (\$49) command from the host for the CGU message contains the range and status parameter.

092: Host generated GRS is missing Range

**Explanation** The host-generated circuit group reset (GRS) ISUP message does not contain the mandatory range parameter.

**Action** Make sure the SS7 Network Message Generation (\$49) command from host for the GRS message contains the range parameter.

093: Circuit is Hardware Blocked

**Explanation** An ISUP unblocking (UBL)/Reset (RSC) message cannot be sent from `isup_console` to clear a hardware-blocked circuit.

**Action** Verify that the port for this circuit is in service in the VCO. If it is not, then the circuit can be cleared from the hardware blocked state by placing it back to in service. If the port is already in service, take the span out-of-service and place it back to the in service state.



094: Group contains Hardware Blocked circuits

**Explanation** The group for which the maintenance message is being sent from isup\_console contains hardware-blocked circuits.

**Action** Verify that the T1 span for this group is in service. If the span is not in service, place it back in service. If it is already in service, unseat and reseal the card.

096: SS7 stack switchover has failed! Trying again...

**Explanation** The SS7 stack switchover has failed.

**Action** Verify that the host is connected. If the host is connected, verify that the fallback switch is operational. If the fallback switch is also operational, contact Cisco Systems TAC.

100: main( ) - No default configuration file available...

**Explanation** The cktint failed to open the cktint.cfg configuration file.

**Action** Verify that the cktint.cfg file is present in the \$XNV directory.

102: main( ) - Error on CALLOC of command line parsing buffers...

**Explanation** The memory allocation failed when reading the cktint.cfg configuration file.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

103: main( ) - SYNTAX error in configuration file...

**Explanation** The cktint detected a syntax error in the CktInt.cfg configuration file.

**Action** Verify that the syntax of the fields in the configuration file is correct. Refer to the appropriate SS7 supplement for instructions.

106: main( ) - Failed trying to setup SIGNALS...

**Explanation** The cktint failed to initialize the signals and exit with a cleanup function.

**Action** Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

107: main( ) - Couldn't get CONTROL shared memory segment...

**Explanation** The cktint could not get the CKTMEM shared memory segment.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

108: main( ) - Couldn't ATTACH to CONTROL shared memory segment pointer...

**Explanation** The cktint failed to attach to the control shared memory segment pointer.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

109: main( ) - Couldn't get CIRCUIT shared memory segment...

**Explanation** The cktint was not able to get the circuit shared memory segment.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

110: main( ) - Couldn't ATTACH to CIRCUIT shared memory segment pointer...

**Explanation** The cktint was not able to attach to the circuit shared memory segment pointer.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

111: main( ) - Couldn't get CIRCUIT GROUP shared memory segment...

**Explanation** The cktint was not able to get the circuit group shared memory segment.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

112: main( ) - Couldn't ATTACH to CIRCUIT GROUP shared memory segment pointer...

**Explanation** The cktint was not able to attach to the circuit group shared memory segment pointer.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

113: main( ) - FAILED on SETUP of ethernet SEMAPHORE...

**Explanation** The cktint failed on the setup of the ethernet semaphore.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

116: main( ) - FAILED on CREATION/OPEN of system message queues...

**Explanation** The cktint failed on the creation or opening of system message queues.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

117: main( ) - FAILED on SPAWN of CHILDREN...

**Explanation** The cktint failed to fork \_ssisan process.

**Action** Verify that no other software is running on the system. Bring down cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

118: main( ) - SpwnChildren FAILED to report in...

**Explanation** The child processes \_ssisan, \_tcprvcvInt are not running. The cktint processes exit.

**Action** Verify that a core file is created. If not, stop cktint and start again.

125: CktExit( ) - FAILED on REMOVAL of semaphore...

**Explanation** The cktint failed to remove a semaphore.

**Action** Stop cktint and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

126: spawnchild( ) - Couldn't FORK

**Explanation** The cktint was not able to create a child process using FORK.

**Action** Stop cktint and the EBS (ADC NewNet) stack and restart. If the problem persists, reboot the system.

127: spawnchild( ) - Couldn't EXEC

**Explanation** Process specified could not be executed.

**Action** Verify that the executable file is present in the \$XNV directory with execute permissions and restart cktint and the EBS (ADC NewNet) stack. If the file is not present, reinstall cktint and restart.

130: ss7\_proc( ) - IAM: invalid parameters: CPN or CHG w/o OLI.

**Explanation** The cktint has received an IAM message from the network with the calling party number (CPN) or charge number (CHG) without originating line information (OLI), which is invalid in ANSI.

**Action** Verify this condition with the network operator. To disable the OLI verification, debug flag 40 should be turned on.

131: `ss7_proc( ) - IAM: invalid parameters: OLI w/o CPN or CHG.`

**Explanation** The `cktint` has received an IAM message from the network with the originating line information (OLI) parameter but without a calling party number (CPN) or charge number (CHG), which is invalid in ANSI.

**Action** Verify this condition with the network operator. To disable the CPN/CHG verification, debug flag 40 should be turned on.

135: `OpenQs( ) - Failed on GET of CKTINT message queue...`

**Explanation** The `cktint` failed to open or create the CKTINT message queue.

**Action** Verify that no other software is running on the system. Bring down `cktint` and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

136: `OpenQs( ) - Failed on GET of HOST message queue...`

**Explanation** The `cktint` failed to open or create the HOST message queue.

**Action** Verify that no other software is running on the system. Bring down `cktint` and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

137: `OpenQs( ) - Failed on GET of SDS message queue...`

**Explanation** The `cktint` failed to open or create the VCO message queue.

**Action** Verify that no other software is running on the system. Bring down `cktint` and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

138: `OpenQs( ) - Failed on GET of SS7 message queue...`

**Explanation** The `cktint` failed to open or create the SS7 message queue.

**Action** Verify that no other software is running on the system. Bring down `cktint` and the EBS (ADC NewNet) stack and restart them. If the problem persists, reboot the system.

140: `catchsigs( ) - ERROR on signal call, SIGTERM`

**Explanation** The signal system call has failed to initialize the SIGTERM signal and exit with a cleanup function.

**Action** Stop `cktint` and the EBS (ADC NewNet) stack and restart. If the problem persists, reboot the system.

141: catchsigs( ) - ERROR on signal call, SIGQUIT

**Explanation** The signal system call has failed to initialize the SIGQUIT signal and exit with a cleanup function.

**Action** Stop cktint and the EBS (ADC NewNet) stack and restart. If the problem persists, reboot the system.

142: catchsigs( ) - ERROR on signal call, SIGINT

**Explanation** The signal system call has failed to initialize the SIGINT signal and exit with a cleanup function.

**Action** Stop cktint and the EBS (ADC NewNet) stack and restart. If the problem persists, reboot the system.

143: catchsigs( ) - ERROR on signal call, SIGHUP

**Explanation** The signal system call has failed to initialize the SIGHUP signal and exit with a cleanup function.

**Action** Stop cktint and the EBS (ADC NewNet) stack and restart. If the problem persists, reboot the system.

144: catchsigs()- ERROR on signal call, SIGUSR1

**Explanation** The signal system call has failed to initialize the SIGUSR1 signal and exit with a cleanup function.

**Action** Stop cktint and the EBS (ADC NewNet) stack and restart. If the problem persists, reboot the system.

145: catchsigs()- ERROR on signal call, SIGCHLD

**Explanation** The signal system call has failed to initialize the SIGCHLD signal and exit with a cleanup function.

**Action** Stop cktint and the EBS (ADC NewNet) stack and restart. If the problem persists, reboot the system.

146: ckt\_config\_at\_startup- could not open resgrp\_file.

**Explanation** The cktint could not open the resource group configuration file res\_grp.cfg.

**Action** Verify that the res\_grp.cfg file exists in the \$XNV directory with the correct data.

148: Failed trying to GET the proper PLTFRMTYP environment variable

**Explanation** The cktint could not get the PLTFRMTYP environment variable which is used to determine whether the system is redundant or standalone.

**Action** Make sure the PLTFRMTYP variable is set to either standalone or redundant, depending upon the requirement found in the .cshrc file. Then stop cktint and EBS. Log out, log in and then restart cktint and the EBS (ADC NewNet) stack.

149: catchsig()- ERROR on signal call, SIGALRM.

**Explanation** The signal system call has failed to initialize the SIGALRM signal and the exit timer handler function.

**Action** Stop cktint and the EBS (ADC NewNet) stack and restart. If the problem persists, reboot the system.