Nortel Media Gateway 7480/15000

Periodic Routine Exercise

NN10600-783

Document status: Standard
Document issue: 7.2S3
Document date: October 2006
Product release: PCR 7.2
Job function: Operations
Type: NTP
Language type: U.S. English

Sourced in Canada and the United States of America.

Contents

New in this release 5

Features 5

REX support for 2pVSP4e FP 5

Other changes 5

Media Gateway 15000 Periodic Routine Exercise overview for the PT-IP GE and UA-IP GE solutions 6

Supported solutions and configurations 7

REX checks and tests 9

Checking Strategy 11

REX provisioning 12

REX test results 14

REX configuration work flow for the PT-IP GE and UA-IP GE solutions 15

REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions 19

Downloading the REX application 22

Configuring the REX feature 23

Scheduling days for REX testing 27

Changing REX start and stop times 29

Scheduling REX to test specific card pairs 30

De-provisioning REX 32

REX operational and manual procedures 34

Starting REX manually 35

Stopping REX manually 36

Disabling REX execution 37

Querying the last REX execution results 38

Querying the current REX activity on the MG15000 40

Accessing the traceLog files produced from REX executions 42

Interpreting REX results 43

Message alarms 44

Alarms 45

Log files 48

Supporting information 67

Shelf configuration checks 68

Card prechecks 69

Postchecks 69

New in this release

The following sections detail what is new in *Nortel Media Gateway 7480/15000 Periodic Routine Exercise (NN10600-783)* for PCR 7.2.

- Features (page 5)
- Other changes (page 5)

Features

See the following section for information about feature changes.

- REX support for 2pVSP4e FP (page 5)

REX support for 2pVSP4e FP

REX support is extended to include the 2pVSP4e FP.

Other changes

There were no other changes made to this document.

Media Gateway 15000 Periodic Routine Exercise overview for the PT-IP GE and UA-IP GE solutions

The MG15000 Periodic Routine Exercise (REX) provides a framework to schedule automated tests on spared hardware equipment, execute those tests, and report results from the tests. This framework consists of the following:

- REX application

  This is the REX package, named rex, that includes the feature and scheduler.

- REX feature

  This is the feature with all of the functionality provided for REX. The REX feature is supported on control processor (CP) cards only and from there invokes its checks and tests on the other supported card types. See Supported solutions and configurations (page 7) to see what is tested.

- REX scheduler

  This is the part of the feature that allows you to schedule REX for the various card pairs, according to your needs, on a daily or weekly basis.

REX is a subsystem that provides periodic routine protected equipment testing on MG15000 nodes. The operator schedules the REX automatic testing of the supported equipment to use component administration system (CAS) provisioning and pre-defined test files that are specific to a selected supported solution. In this release, only PT-IP GE or UA-IP GE solutions are supported. Operator configured parameters specify the set of cards to be tested, at a specific time, on a given day of the week. The operator can also run REX manually when the automated REX executions are not scheduled.

It is recommended that you configure REX to run a routine exercise on spared equipment to ensure that standby equipment is ready to take activity. Schedule the testing to occur at off-peak hours. REX automatically tests the spared card of a pair to guarantee that both units of a pair are healthy and to ensure that the spare unit is ready and able to provide full service in the case where a service-providing unit experiences a failure.

Manual starting and stopping of REX is also supported to allow the operator the ability to invoke REX tests directly on a specified card pair at any time using operator commands from a command line interface on the MDM. You can run REX manually on specified spared cards provided the automated REX tests are not in progress. REX automated testing and manual testing cannot be run at the same time. If REX is performing a scheduled automated test and a manual test is invoked, the manual REX test will fail.

Attention: By default REX is ON and runs automatically every day at 02:00 on all spared cards and continue until 04:30. The tests abort if they do not complete by 04:30.

Supported solutions and configurations

REX tests are implemented on a Carrier VoIP Media Gateway 15000 platform that uses specific Voice Service Processors function processors (FP). The Carrier VoIP solutions that support the MG15000 REX application are as follows:

- Packet Trunking-IP Gigabit Ethernet (PT-IP GE)
  - allows an InterExchange (IXC) tandem, or Local Exchange Carrier (LEC) local tandem to migrate an existing TDM based tandem/toll switch to packet technology
- Universal Access-IP Gigabit Ethernet (UA-IP GE)
  - allows a Local Exchange Carrier (LEC) to migrate an existing end-office/local TDM switch to packet technology

Attention: The REX feature is not supported for any PT-IP GE or UA-IP GE configurations that contain ATM cards.

REX is supported on the following equipment provided the configuration meets the requirements described in this section:

- MG15000 shelf
- CP3 (DS1 BITS CP (NTHW06) or E1 BITS CP (NTHW08)
- Fabric cards (NTHR16)
- 4-port gigabit Ethernet FP card (4pGigE (NTHW49))
- VSP3-o FP card (2pOC3ChSmIrVsp3 (NTHW77))

- 2pVSP4e FP card (2pOc3ChSmIrVsp4e (NTHW26))

Attention: Any cards that have been UP (in-service) for less than 24 hours are not eligible for automatic REX testing.

REX verifies the health of protected cards in the MG15000 shelf thereby providing a proactive mechanism to help prevent total and partial service outages.

Attention: REX tests are run on the spared card of the pair. For the duration of the time it takes to test the spared card, your active card will be unprotected.

Failures on an active card usually cause a protection switch to a mate card in the supported protection schemes. If the mate card has a problem that inhibits a switch to protection, a total service outage can result as explained in the following list. REX tests can be run on equipment pairs with the following protection schemes on an MG15000 node deployed in a Carrier VoIP solution:

- 1:1 CPs which operate in hot standby mode.

  Failure of both CPs leads to a total service outage; card slot location is 0/1.

- 1+1 fabrics which operate in a load sharing mode

  Failure of both fabrics leads to a total service outage; fabrics are labeled X/Y.

- 1:1 FPs which operate in hot standby mode

  Failure of both FPs in a protected pair leads to a partial service outage; card slot locations of protected FP pairs are 2/3, 4/5, 6/7, 8/9, 10/11, 12/13, and 14/15.

- 1+1 protection of MS3 4-port GE FPs that operate in a load sharing mode

  Failure of both FPs leads to a total service outage for all calls going off or coming into the MG15000 shelf.

Attention: The REX application only supports 1+1 and 1:1 equipment protection. Unprotected and 1:N protection schemes are not supported and must not appear in the card pair list for REX execution.

Attention: Ensure that REX is not scheduled to run at the same time as REX on other devices that are connected to the MG15000 where REX is scheduled to run. Do not schedule REX to run on adjacent network nodes at the same time.

REX checks and tests

The REX feature runs CAS commands automatically to perform the following checks and tests as part of its execution:

- Configuration prechecks
- Shelf and card prechecks
- REX diagnostic card tests
- shelf and card postchecks

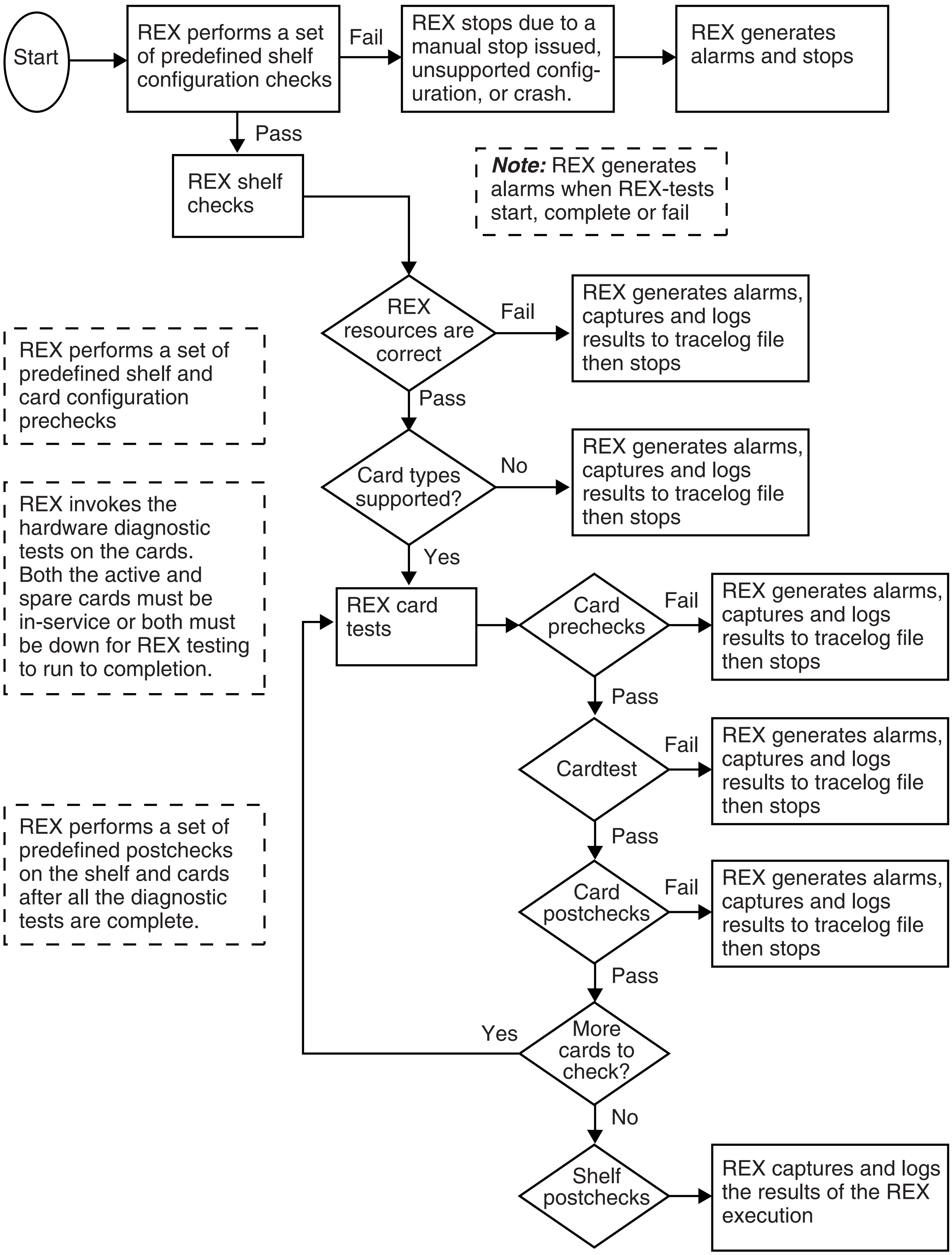
REX tests can be scheduled ON or OFF for each day of the week and for each card pair. Provided that all of the prechecks complete successfully, the REX diagnostic tests are run to verify the health of the cards specified. You can let REX run automatically on all the cards on the shelf or run REX on cards that you specify. The results are parsed and interpreted against a pass/fail criteria which are then passed back to the REX feature as a pass or fail for further processing such as raising alarms upon failure.

Shelf postchecks are run after the REX card tests have finished and include a replay of the prechecks with a check of the active alarm list to ensure that no new alarms are present after REX execution.

To ensure that the REX test completes successfully, a comparison is made of the active alarms before and after the REX application has run to determine if any additional alarms were introduced during the REX run. If additional alarms are detected in the shelf postchecks, the REX application reports a failure due to the detection of additional alarms.

Finally, REX generates logs and alarms for every part of the test execution, see the figure Logical flow of REX-testing (page 10) to understand the REX testing process.

Logical flow of REX-testing


Start
REX performs a set of predefined shelf configuration checks
Fail
REX stops due to a manual stop issued, unsupported configuration, or crash.
REX generates alarms and stops
Pass
REX shelf checks
Note: REX generates alarms when REX-tests start, complete or fail
REX resources are correct
Fail
REX generates alarms, captures and logs results to tracelog file then stops
Pass
REX performs a set of predefined shelf and card configuration prechecks
Card types supported?
No
REX generates alarms, captures and logs results to tracelog file then stops
Yes
REX invokes the hardware diagnostic tests on the cards. Both the active and spare cards must be in-service or both must be down for REX testing to run to completion.
REX card tests
Card prechecks
Fail
REX generates alarms, captures and logs results to tracelog file then stops
Pass
Cardtest
Fail
REX generates alarms, captures and logs results to tracelog file then stops
Pass
REX performs a set of predefined postchecks on the shelf and cards after all the diagnostic tests are complete.
Card postchecks
Fail
REX generates alarms, captures and logs results to tracelog file then stops
Pass
Yes
More cards to check?
No
Shelf postchecks
REX captures and logs the results of the REX execution

Checking Strategy

Before any hardware testing (including REX diagnostic testing), REX triggers a shelf-wide configuration test and a series of prechecks to determine the relative health of the shelf. If conditions are found where running REX may result in a service outage (for example, loss of redundancy or MG15000 congestion), alarms are raised and REX stops running. Once the shelf-wide tests are complete, the health of the individual card pairs selected for REX testing are examined. For REX to run to completion, active and standby cards must be either both in-service or both down. After REX has run to completion, postchecks are made to ensure the shelf and cards are in a healthy state.

The strategy is to perform enough pre-testing at each level to determine that the shelf and cards are in a healthy state throughout the REX testing process. A set of postchecks ensure that no new problems have resulted from the REX process. The process is as follows:

- The shelf configuration is prechecked to determine the system’s health and if there is a problem, alarms are raised and REX stops.
- Then the shelf and card configurations are prechecked to determine their health prior to REX-testing. If there is a problem, alarms are raised and REX stops.
- Then the specific spared cards are tested and if there is a problem, alarms are raised and REX stops.
- Then the prechecks are replayed as postchecks.
- Finally, logs are written each time that REX passes a test or detects a problem and fails. The log files and alarms raised for each issue assists you in troubleshooting and fixing all the problems that must be resolved.

REX performs one set of shelf tests to verify that the shelf meets a certain minimum criteria of conditions to allow the REX application to start individual card testing. A shelf-level check ensures the MG15000 is in a reasonable system-wide state for card level checking and testing to start. The checks are the same regardless of what card types are selected for testing. For REX to run to completion, active and standby cards must be either both in-service or both down.

Before REX testing can proceed, a check is made to ensure that the sparing status is functioning correctly and to ensure that running REX tests on a service that is down will not make service impairment any worse. This also allows for the case where the service has been commissioned and provisioned on the MG15000 shelf, but the far-end node is not yet available.

Before REX finishes its run on the first card pair, it checks if there are other card pairs to be tested and cycles through the specified card prechecks and testing for each of those pairs. After REX completes testing, postchecks run to ensure that the cards and shelf are healthy and ready to carry traffic. Finally, logs are written that capture all the activity encountered during the REX execution.

REX provisioning

Initial REX provisioning is performed with the Nodal Provisioning Template SH-REX-CardPair or SH-REX-Schedule (according to the solution) at start up. Initial provisioning includes:

- downloading the REX application
- provisioning the REX application
- provisioning the REX feature
- scheduling REX to run at specific times on specific card pairs

The initial configuration has REX running from the time the feature is provisioned using the default configuration. By default, REX starts everyday, testing every card on the shelf, at 02:00 and stops everyday at 04:30.

Scheduling REX to run at specific times on specific card pairs is part of the initial provisioning for the feature. You can change the REX schedule at any time after the feature has been configured.

Change the default configuration at any time to suit your scheduling requirements using the scheduling procedures for the feature. Weekly scheduling changes that coincide with maintenance windows are recommended but scheduling can be done daily. See the following to understand REX provisioning and scheduling:

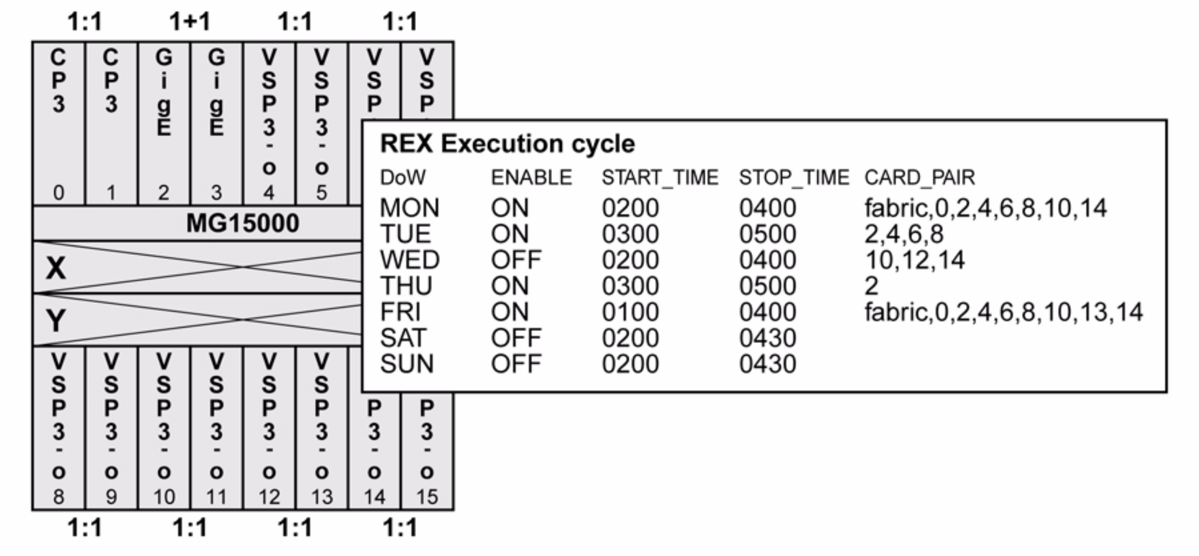
- REX configuration work flow for the PT-IP GE and UA-IP GE solutions (page 15)
- REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions (page 19)

The figure MG15000 shelf REX provisioning concept (page 13) is a conceptual representation of REX scheduling after it has been provisioned and the default scheduling has been changed. The figure includes:

- an MG15000 card fill
- an Execution cycle

The user-defined input schedule data that is required for the REX application is labelled “Execution cycle”. That is, you configure the values as seen in the Execution cycle for the days testing is to occur. In the figure, the component attribute ENABLED, ON or OFF, with configured start and stop times, and configured card pairs to be tested indicate whether REX runs for that day.

MG15000 shelf REX provisioning concept



REX Execution cycle

DoW	ENABLE	START_TIME	STOP_TIME	CARD_PAIR
MON	ON	0200	0400	fabric,0,2,4,6,8,10,14
TUE	ON	0300	0500	2,4,6,8
WED	OFF	0200	0400	10,12,14
THU	ON	0300	0500	2
FRI	ON	0100	0400	fabric,0,2,4,6,8,10,13,14
SAT	OFF	0200	0430	
SUN	OFF	0200	0430	

In the execution cycle, the columns in the example refer to what has been scheduled. The columns are as follows:

- DoW refers to the day of the week that REX is intended to execute. Each day of the week has an associated active state, start time, stop time, and card pair(s) that the REX application will run on for the particular day.
- ENABLE indicates whether or not REX executes on the given night. By default, REX is scheduled to run every night (ENABLE=ON). You must provision REX to skip (ENABLE=OFF) any day that you do not want REX to run. The configured values for the ENABLE attribute are YES for ON and NO for OFF. This functionality allows you to disable REX testing when necessary but maintain the weekly schedule for maintenance shifts up to a week ahead of time and allow for such activities as software migration.
- START_TIME refers to the time at which REX starts its test execution cycle. The units are <hours>:<minutes> using the 24 hour clock (00:05 to 23:58). The earliest start time for REX testing is 00:05. In the example, on Monday, REX is to start at 02:00. On Saturday and Sunday there are no tests scheduled. The default value for START_TIME is 02:00.
- STOP_TIME refers to the time at which REX is required to stop its execution. The units are <hours>:<minutes> using the 24 hour clock (00:06 to 23:59). REX will stop at the time configured regardless if it has completed all scheduled tests or not. STOP_TIME cannot be provisioned as less than START_TIME. This means that the stop time must be within the same 24 hour period, For example, you cannot have a start time for a REX run at 23:30 and a stop time for the REX run at 00:30. Card pair tests are performed sequentially, not in parallel.

- CARD_PAIR refers to the even numbered slot of the protected card pair and is considered independent of which card is active. For Example, the CP pair in slots 0 and 1 is represented with a 0. The Fabric does not have a slot number in the front of the shelf and is referred to as fabric. The FP pairs are represented as 2 (for slots 2 and 3), 4 (for slots 4 and 5) and so on.

REX test results

REX-tests produce alarms and logs which can be used to proactively troubleshoot the MG15000 shelf and supported spared cards for their ability to carry traffic in the case where a switchover to protection is required. Your careful analysis of the alarms and logs generated from REX testing gives you insight as to the general health of the MG15000 and its card pairs for the configurations on which REX is supported. See Interpreting REX results (page 43) for a description of REX alarms and logs.

REX configuration work flow for the PT-IP GE and UA-IP GE solutions

The Periodic Routine Exercise (REX) configuration task flow illustrates the sequence of tasks that you perform to configure REX-testing in your Carrier VoIP PT-IP GE or UA-IP GE solutions. See the Periodic Routine Exercise (REX) configuration task flow (page 17). Each box in the task flow represents a task that comprises one or more procedures. Each task has a corresponding section in this guide that contains the relevant procedures. To link to any task, go to the Navigation links (page 18).

The REX feature is enabled and the REX scheduler runs the automatic hardware tests routinely every day from the moment the application is downloaded and the feature is provisioned. You can customize the default configuration using the procedures in REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions (page 19).

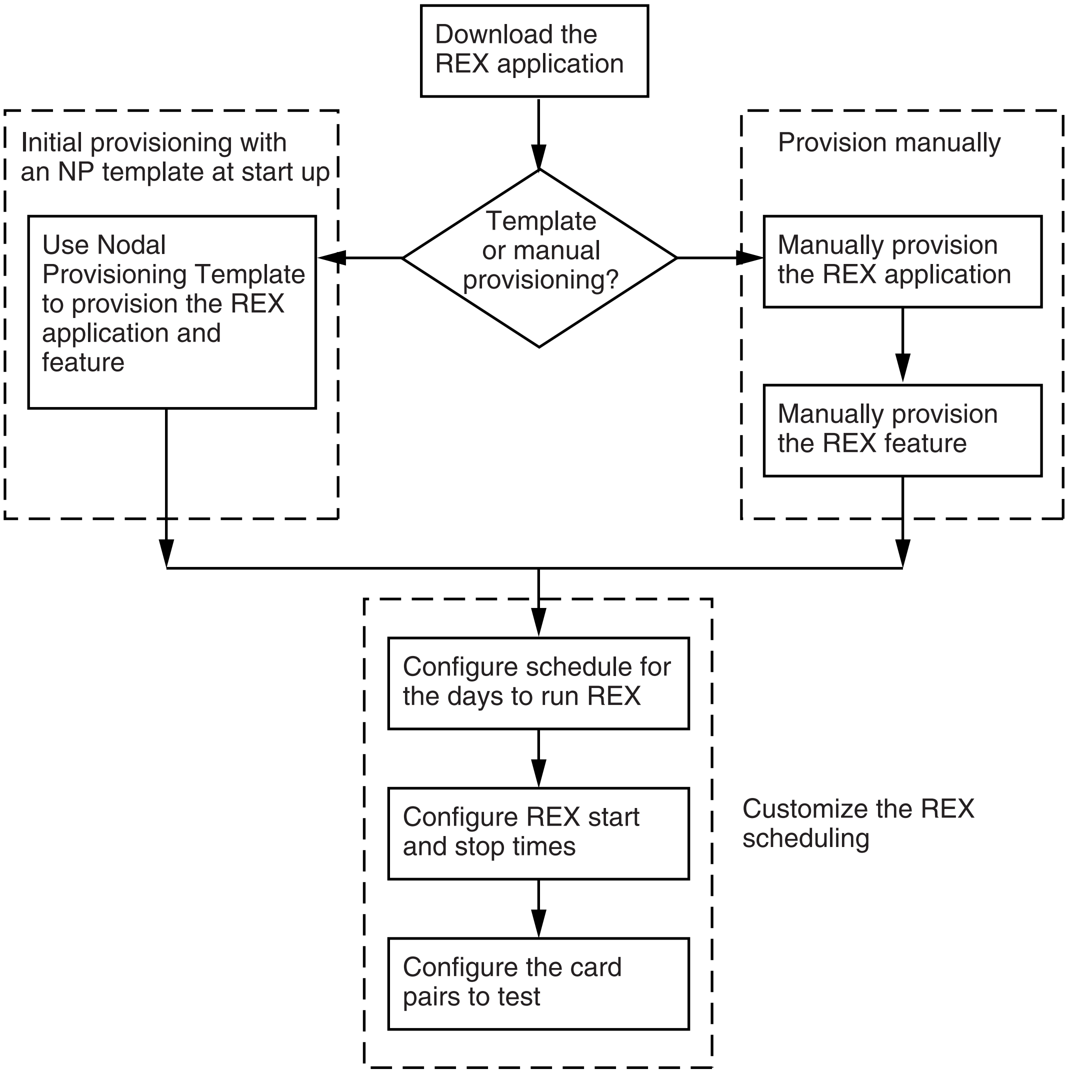
REX configuration prerequisites

- Ensure you have the software application named REX.
- Schedule REX to run during maintenance windows.
- Ensure you have connectivity to the MDM.
- Ensure the network is configured for PT-IP GE or UA-IP GE only, with no ATM.
- Ensure the shelf is configured with only the supported cards in supported configurations with no ATM.
- Ensure that REX is not scheduled to run at the same time as a REX application on other devices in the network vicinity or that are connected to the MG15000 far-end node as this causes simultaneous testing of protected equipment which can result in a service interruption.
- After you have provisioned the REX feature, you must configure REX scheduling for start times, stop times, and days of the week on which you want to run REX.
- Ensure the cards to be tested have been in the shelf, provisioned, and UP for more than 24 hours.

Periodic Routine Exercise (REX) configuration task flow

The following task flow shows the sequence of procedures required to obtain and configure REX into your Carrier VoIP PT-IP GE or UA-IP GE solution. To link to any procedure, go to Navigation links (page 18).

REX configuration task flow


Download the REX application
Template or manual provisioning?
Initial provisioning with an NP template at start up
Use Nodal Provisioning Template to provision the REX application and feature
Provision manually
Manually provision the REX application
Manually provision the REX feature
Configure schedule for the days to run REX
Configure REX start and stop times
Configure the card pairs to test
Customize the REX scheduling

Navigation links

- REX provisioning task flow (page 20)
- Downloading the REX application (page 22)

  If you are going to use the Nodal Provisioning Template SH-REX-CardPair or SH-REX-Schedule to provision the REX application and feature, refer to *Nortel Multiservice Data Manager Administration—Tools* (NN10470-300) for information about using the tool and to *NN10225-512, Nortel Multiservice Switch 15000, Media Gateway 15000 and Multiservice Data Manager in Carrier Voice over IP Networks Configuration Attribute Summary (PT-AAL1/UA-AAL1/UA-IP/PT-AAL2)* for the component attributes.
- Configuring the REX feature (page 23)
- Scheduling days for REX testing (page 27)
- Changing REX start and stop times (page 29)
- Scheduling REX to test specific card pairs (page 30)
- De-provisioning REX (page 32)

REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions

Configure the Media Gateway 15000 Periodic Routine Exercise (REX) on the control processor (CP) to perform scheduled automated testing of spared cards that operate in the Carrier VoIP PT-IP GE and UA-IP GE solutions. The application is delivered in a software application named REX.

The REX application contains the CDL software that corresponds to the REX data model and must be downloaded onto the switch before the REX feature can be provisioned. The REX feature contains the software to schedule and execute hardware tests on the supported spared hardware units.

Initial provisioning of the REX application and feature can be done using a Nodal Provisioning Template available from the Nodal Provisioning Tool on the MDM or it can be done manually as is described in this chapter. This chapter provides steps to manually provision the REX application and feature and to customize the default REX schedule.

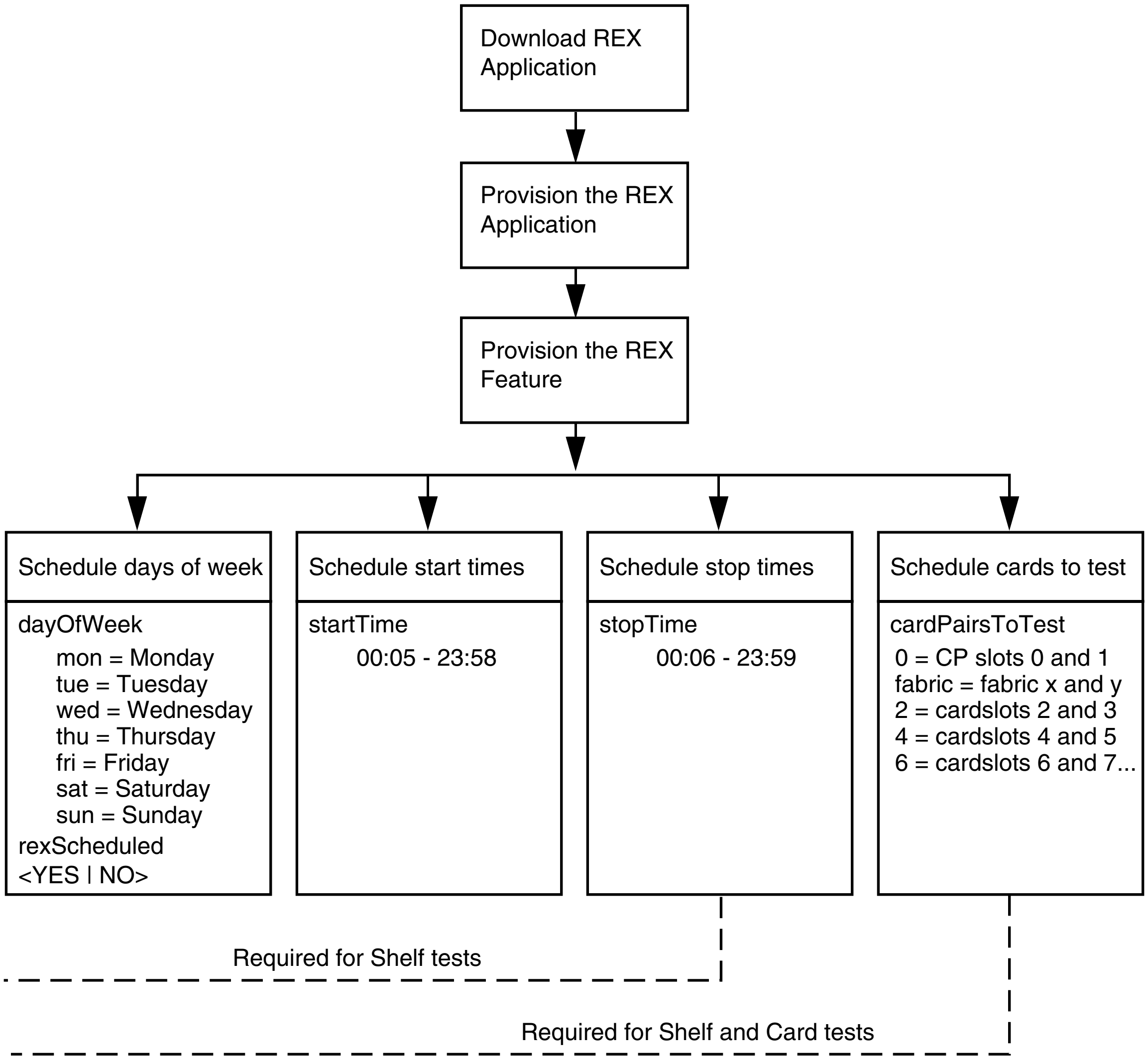
Prerequisites

- You must have the supported cards in the supported configurations on the shelf. See Supported solutions and configurations (page 7).
- Ensure that ATM is not provisioned on the same shelf.
- Ensure that the cards to be tested have been provisioned in the shelf and in an UP state for at least 24 hours.

REX provisioning task flow

The following task flow shows the sequence of procedures to configure REX testing for the shelf and card in your Carrier VoIP PT-IP GE or UA-IP GE solution. To link to any procedure, go to Navigation (page 20).

REX provisioning task flow



Navigation

- Downloading the REX application (page 22)

  If you are going to use the Nodal Provisioning Template SH-REX-CardPair and SH-REX-Schedule to provision the REX application and feature, refer to *Nortel Multiservice Data Manager Administration—Tools*

(NN10470-300) for information about using the tool and to *NN10225-512, Nortel Multiservice Switch 15000, Media Gateway 15000 and Multiservice Data Manager in Carrier Voice over IP Networks Configuration Attribute Summary (PT-AAL1/UA-AAL1/UA-IP/PT-AAL2)* for the component attributes.

- Configuring the REX feature (page 23)
- Scheduling days for REX testing (page 27)
- Changing REX start and stop times (page 29)
- Scheduling REX to test specific card pairs (page 30)
- De-provisioning REX (page 32)

Downloading the REX application

For more information on downloading applications, refer to *Nortel Multiservice Switch 7400/15000/20000 Installation – Software* (NN10600-270).

Configuring the REX feature

Configure the REX feature on the switch manually at startup. The preferred and recommended method to provision this feature is to use the Nodal Provisioning Template available from the MDM. Refer to NN10225-512 *Nortel Multiservice Switch 15000, Media Gateway 15000 and Multiservice Data Manager in Carrier Voice over IP Networks Configuration Attribute Summary (PT-AAL1/UA-AAL1/UA-IP/PT-AAL2)* for more information about REX provisioned attributes.

By default, the REX feature is enabled and the REX scheduler runs to invoke the automatic hardware tests routinely every day from the moment the application is downloaded and provisioned. REX immediately begins testing at its default time using its default configuration (start time 02:00 and finish time 04:30). After you have provisioned the REX feature, you must configure REX scheduling for days of the week on which you want to run REX testing, start times and stop times, and card pairs to test.

Prerequisites

Attention: This procedure is not required if you are performing an initial configuration of REX using the Nodal Provisioning Template at start up.

- Ensure you have enough disk space for the REX application.
- Ensure you have downloaded the application from the Nortel download site.
- Ensure you have access with systemAdmin impact to the Media Gateway 15000.
- Ensure that no other provisioning activity is currently in progress.
- Ensure that REX is not scheduled to run at the same time as REX on other devices that are connected to the MG15000 where REX is scheduled to run.
- Ensure the cards have been provisioned in the shelf and in an UP state for at least 24 hours.
- Ensure you know the correct syntax for the provisioning commands. For more information, refer to *Nortel Multiservice Switch 7400/15000/20000 Commands Reference* (NN10600-050).

Procedure steps

Step	Action
1	Log in to the Media Gateway 15000.

2 Issue the following commands:

```
start prov
PROV> set sw avl REX_<version>
PROV> set sw lpt/<CP_lptName> featureList rex
PROV> check prov
PROV> save prov
PROV> activate prov
PROV> confirm prov
```

3 Add the REX component.

```
start prov
PROV> add rex
PROV> check prov
PROV> activate prov
PROV> save prov
PROV> confirm prov
PROV> commit prov
```

4 Configure REX scheduling in provisioning mode.

5 Schedule REX to run on one shelf at a time and ensure that no other shelf in the network vicinity is scheduled to run REX at the same time. For example, ensure that REX is not scheduled to run at the same time as far-end nodes.

```
PROV> d rex
enable = YES
solutionType = <ptIPGE|uaIPGE>
```

The solutionType attribute value determines the set of pre-check and post-check commands to run.

```
PROV> d rex dayOfWeek/<day of week>
```

The default value for day of the week is sun for Sunday.

Attention: REX will only run on cards that have been UP for more than 24 hours.

```
rexScheduled = <YES | NO>
```

YES is the default setting.

```
startTime = <00:05 to 23:58>
```

02:00 is the default setting

```
stopTime = <00:06 to 23:59>
```

04:30 is the default setting

```
cardPairsToTest = <fabric or even numbered slot of each
pair>
```

Blank is the default value. As an example: (0 fabric 2 4 6...)

Attention: The attribute cardPairToTest uses bit set to indicate if the pair of protected cards is scheduled to run REX on the configured day. The cardPairsToTest attribute uses 0 to represent the first pair of cards (CPs on MSS 15000) in slot 0 and 1, fabric, then 2 to represent the second pair of cards in slot 2 and 3, and so on.

Attention: For load-sharing cards such as fabric cards and 4pGigE FPs, card slot numbers are assigned and the Rex scheduler alternates between the cards. That is, it tests one card on one day and the other card the next day.

Attention: Ensure that you schedule at least 24 hours between tests for REX runs that are to occur on the same card. If REX is scheduled to run on a card that has been tested within the same 24 hour period, REX will fail.

--End--

Variable definitions for REX provisioning

Variable	Definition
sw lpt	<CP_lptName> Identifies the active CP.
enable	<Yes | No> Yes enables REX testing.
solutionType	<PT-IP GE | UA-IP GE | NONE> The solutionType attribute value determines the set of pre-check and post-check commands to run according to the solution.

dayOfWeek	<day of week>. REX configuration data for each day of the week. Sunday is considered the first day of the week.
	mon = Monday
	tue = Tuesday
	wed = Wednesday
	thu = Thursday
	fri = Friday
	sat = Saturday
	sun = Sunday

rexScheduled	<YES | NO> YES is the default setting.

Variable	Definition
startTime	<00:05 to 23:58> The startTime for a REX run cannot be before 00:05. REX is scheduled to start at 02:00 by default.
stopTime	<00:06 to 23:59> The default stopTime for REX is 04:30.
cardPairsToTest	<0, fabric, 2, 4, ...> 0 represents the first pair of cards (CPs on MSS 15000) in slot 0 and 1. fabric represents the fabrics x and y. 2 represents the second pair of cards in slot 2 and 3. 4 represents the third pair of cards in slot 4 and 5, and so on.

Scheduling days for REX testing

By default, REX is turned ON for each day in the entire week. To suspend or skip the REX tests on a specific day, you must set the rexScheduled attribute that corresponds to the DayOfWeek component instance to NO. To re-enable REX tests for the same day in the subsequent week, you must set the rexScheduled attribute value to YES for that day of the week when you want REX to run.

Prerequisites

- Ensure you have access with systemAdmin impact to the Media Gateway 15000.
- Ensure that no other provisioning activity is currently in progress.
- Ensure that REX is not scheduled to run at the same time as REX on other devices that are connected to the MG15000 where REX is scheduled to run.
- Ensure you know the correct syntax for the provisioning commands. For more information, refer to *Nortel Multiservice Switch 7400/15000/20000 Commands Reference* (NN10600-050).

Procedure steps

Step **Action**

1 Log in to the Media Gateway 15000.

2 Issue the following commands:

```
start prov
PROV> set rex dayOfWeek/<day of week> rexScheduled NO
```

Attention: <day of week> is the dayOfWeek value (mon, tue, wed, ...). When the rexScheduled component value is set to NO on the configured day of the week, the REX application is scheduled to not test for that day. The rexScheduled component can be configured as YES or NO for the day.

```
PROV> check prov
PROV> activate prov
PROV> save prov
PROV> confirm prov
PROV> commit prov
```

Attention: To enable REX testing for the same day in the next week, verify that the schedule for the current week has expired and then go to step 3.

3 Set the rexScheduled attribute value to YES to re-enable REX tests for the same day in the subsequent week.

```
start prov
PROV> set rex dayOfWeek/<day of week> rexScheduled YES
PROV> check prov
PROV> activate prov
PROV> save prov
PROV> confirm prov
PROV> commit prov
```

--End--

Changing REX start and stop times

Change REX start and stop times from their previously configured times. REX tests are scheduled on or off for each day of the week to start and stop at certain times. The start time and/or stop time for REX automatic tests can be configured on a daily basis. It is recommended that the REX feature be scheduled to run during maintenance windows.

Prerequisites

- Ensure you have access with systemAdmin impact to the Media Gateway 15000.
- Ensure that no other provisioning activity is currently in progress.
- Ensure that REX is not scheduled to run at the same time as REX on other devices that are connected to the MG15000 where REX is scheduled to run.
- Ensure you know the correct syntax for the provisioning commands. For more information, refer to *Nortel Multiservice Switch 7400/15000/20000 Commands Reference* (NN10600-050).

Procedure steps

Step	Action
1	Log in to the Media Gateway 15000.
2	Issue the following commands:

```
start prov
PROV> set rex dayOfWeek/<day of week> startTime <00:00
to 23:59>
PROV> set rex dayOfWeek/<day of week> stopTime <00:00 to
23:59>
```

Attention: <day of week> is the dayOfWeek value on which your changed REX start time and/or stop time are configured.

```
PROV> check prov
PROV> activate prov
PROV> save prov
PROV> confirm prov
PROV> commit prov
```

--End--

Scheduling REX to test specific card pairs

Schedule specific card pairs for REX testing. If you do not configure the cardPairsToTest attribute, by default REX runs the shelf prechecks only.

Prerequisites

- Ensure you have access with systemAdmin impact to the Media Gateway 15000.
- Ensure that no other provisioning activity is currently in progress.
- Ensure that REX is not scheduled to run at the same time as REX on other devices that are connected to the MG15000 where REX is scheduled to run.
- Ensure the cards have been provisioned in the shelf and in an UP state for at least 24 hours.
- Ensure you know the correct syntax for the provisioning commands. For more information, refer to *Nortel Multiservice Switch 7400/15000/20000 Commands Reference* (NN10600-050).

Procedure steps

Step **Action**

1 Log in to the Media Gateway 15000.

2 Issue the following commands:

```
start prov
PROV>set rex dayOfWeek/<day of week> cardPairsToTest
<card pairs>
```

Attention: <day of week> stands for the dayOfWeek value when REX is scheduled to run for a subset of the card pairs.

```
PROV> check prov
PROV> activate prov
PROV> save prov
PROV> confirm prov
PROV> commit prov
```

--End--

Variable definitions for cardPairsToTest attribute

Variable	Definition
`<card pairs>`	fabric - both fabric cards x and y 0 - the first card pair for the CPs in slots 0 and 1 2, 4, 6, ...14 - the single value that represents active and standby FPs in two slots (i.e., 2 represents the FP card pairs which are the FPs in slots 2 and 3)

De-provisioning REX

This procedure de-provisions REX completely. If you decide that you want REX testing after you have performed this procedure, you will need to perform the procedures required for Configuring the REX feature (page 23).

It is recommended that rather than de-provisioning REX, you disable REX. For more information, refer to Disabling REX execution (page 37).

> **Traffic outage**
>
> Rex is a CP-based feature. If you de-provision REX, this may cause a CP reset.

Prerequisites

- Ensure you have access with systemAdmin impact to the Media Gateway 15000.
- Ensure that no other provisioning activity is currently in progress.

Procedure steps

Step **Action**

1 Log in to the Media Gateway 15000.

2 Issue the following commands:

```
start prov
PROV> delete rex
```

3 The following commands are optional and will result in a system reload.

```
PROV> set sw lpt/<CP_lptName> featureList ~rex
PROV> set sw avl ~rex_<version>
PROV> check prov
```

Attention: REX is a CP feature, the check prov output will indicate that removing the feature causes a system reload.

```
PROV> activate prov
```

Attention: The MSS 15000 will experience a system reload which causes total traffic outage.

```
PROV> confirm prov
PROV> commit prov
```

--End--

REX operational and manual procedures

Manual control of REX testing is available with this feature. In addition to scheduled automated REX testing, this feature supports manual starts for REX testing. That is, you can start REX outside of the scheduled runs and the system will carry out the same tests on selected cards. The manual start of REX tests can be performed, as long as no scheduled tests are in progress. You can perform the following manual operations:

- Starting REX manually (page 35)
- Stopping REX manually (page 36)
- Disabling REX execution (page 37)
- Querying the last REX execution results (page 38)
- Querying the current REX activity on the MG15000 (page 40)
- Accessing the traceLog files produced from REX executions (page 42)

Starting REX manually

You can start the same hardware tests on specified spared cards manually provided that automatic testing is not running. Set the parameter <specified spared card> to the card that you want to test.

Prerequisites

- Ensure REX has been provisioned according to REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions (page 19).
- You must have the supported cards in the supported configurations on the shelf with no ATM.
- Ensure you have access with systemAdmin impact to the Media Gateway 15000.
- Ensure that scheduled automated REX testing is not in progress.
- Ensure the cards to be tested have been provisioned in the shelf and in an UP state for more than 24 hours.

Procedure steps to start REX manually

Step	Action
1	Log in to the Media Gateway 15000.
2	Issue the following command: Note that when you issue the “start rex” command with no parameters (blank), REX performs the shelf-wide prechecks only. `start -card(<specified spared cards>) rex`

--End--

Variable definitions for the start rex command

Variable	Definition
<specified spared cards>	<blank | fabric | slot number> where slot number is the slot number of the card you want to test and fabric is fabricX or fabricY. The card prechecks and hardware tests are executed after the shelf-wide prechecks for the specified card.

Stopping REX manually

You can abort the REX test at any time. If the stop command is issued in the middle of the REX execution, REX execution is stopped after completing the tests on the current card. If the shelf wide prechecks or postchecks are in progress on a scheduled REX run or manual REX start, the REX execution aborts when it completes the entire set of checks. For all runs, REX exits and goes to the STOPPING state when finished. A major message alarm is raised to indicate the transition.

Prerequisites

- Ensure REX has been provisioned according to REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions (page 19).

Procedure steps to stop REX manually

Step	Action
1	Log in to the Media Gateway 15000.
2	Issue the following command: `stop rex`

Attention: If REX was manually aborted, which puts REX in a STOPPING state, automated REX testing resumes at the next scheduled interval if the ENABLE attribute under the **rex** component is provisioned YES and the next **DayOfWeek rexScheduled** attribute is provisioned YES. If the next interval is in less than 24 hours, the same card will not be tested.

--End--

Disabling REX execution

You can disable the REX component completely without de-provisioning it. The scheduled automated REX testing is disabled with this procedure. However, the manual REX testing is not disabled.

To make the REX feature active again you must turn it on manually. A minor set/clear alarm is raised daily as long as REX is disabled to remind you that the feature is disabled. The alarm indicates the current number of days that REX has been disabled.

Prerequisites

- Ensure REX has been provisioned according to REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions (page 19).
- Ensure that scheduled automated REX testing is not in progress.

Procedure steps to disable REX manually

Step **Action**

1 Log in to the Media Gateway 15000.

2 Issue the following command:

```
start prov
PROV> set rex enable NO
PROV> check prov
PROV> activate prov
PROV> confirm prov
PROV> commit prov
```

When rex enabled is set to "NO", REX scheduling for automatic test is disabled. However, manual REX test can still start.

--End--

Querying the last REX execution results

You can query the last REX execution results with possible attribute values using this procedure.

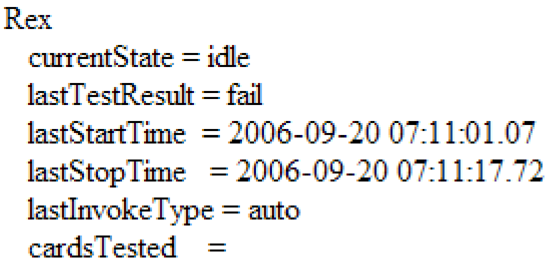
Prerequisites

- Ensure REX has been provisioned according to REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions (page 19).
- You must have the supported cards in the supported configurations on the shelf with no ATM.

Procedure steps to query the results of the last REX tests

Step	Action
1	Log in to the Media Gateway 15000.
2	Issue one of the following commands: To display the test results from the last REX execution cycle, enter: **`d rex lastTestResult`** To display the test results from the last REX execution cycle as well as the weekly schedule, enter: **`query rex`** An example of the response from these commands is shown in Response for d rex (page 38) and Response for query rex (page 39).

Response for d rex



```
Rex
  currentState = idle
  lastTestResult = fail
  lastStartTime  = 2006-09-20 07:11:01.07
  lastStopTime   = 2006-09-20 07:11:17.72
  lastInvokeType = auto
  cardsTested    =
```

Response for query rex

```
Rex
================================================
LAST TEST RESULTS
Last Test Result: Fail
Last Start Time: 2005-07-20 14:46:16
Last Stop Time : 2005-07-20 14:46:33
Type Of Test: Manual
Cards Tested:
================================================
================================================
| DAY OF WEEK DETAILS |
|                     |
|SOLUTION TYPE : UA_IPGe|
+---+-------------+----------+---------+-------------------+--
|Dow|rexScheduled startTime stopTime |cardPairToTest
+---+-------------+----------+---------+-------------------+--
|Sun| No           |         |         |                 |
|Mon| Yes          | 11:29   | 16:30   | 0,12,14         |
|Tue| Yes          | 12:45   | 18:20   | fabric,8,10     |
|Wed| Yes          | 02:00   | 04:30   | 0,2,4,6,12      |
|Thu| Yes          | 02:00   | 04:30   |                 |
|Fri| Yes          | 02:00   | 04:30   |                 |
|Sat| Yes          | 02:00   | 04:30                     |
+---+------------|+---------|+---------+-----------------|------
ok 2005-07-20 14:51:24.94
```

--End--

Querying the current REX activity on the MG15000

You can check the current REX activity on switch with this procedure. The response indicates the current state of REX activity on the switch. If REX is disabled, the value of the currentState attribute shows UNINITIALIZED. If REX functionality is enabled but it is not time to invoke the REX testing, the currentState attribute shows IDLE. If the operator stops REX in the middle of test execution, the currentState attribute shows STOPPING. The state transitions that REX experiences as it executes and explanations of what is happening with each state change follows:

- UNINITIALIZED

  If REX is disabled, REX is in an UNINITIALIZED state. When the REX feature is enabled and the application moves to the IDLE state.

- IDLE

  REX is enabled and waiting for the clock to start the next cycle of tests.

- STOPPING

  REX stops when it encounters errors or has been stopped manually. If REX is STOPPING, a message alarm is raised and the REX application sets its timer for the next scheduled period.

- PRECHECK

  As part of the prechecks, REX checks if the configuration data is correct, such as having the correct solution precheck file available. A message alarm is raised if the REX application cannot find all the resource files it needs to successfully execute. If the configuration data is correct, REX generates a minor set alarm for REX start, opens the traceLog file and moves its state to TESTING_CARD.

  If the prechecks fail, a message alarm is raised to indicate that the shelf prechecks have failed and the traceLog has the details of the failures. A Clear alarm is raised to indicate that the REX application has completed and the state is changed to IDLE. If the prechecks are successful, then the results are written to the traceFile and REX prepares for the individual card tests.

- TESTING_CARD<n> (where n stands for the card number)

  When the shelf prechecks are complete, a series of checks are made to determine if the card type selected for REX test is supported. If yes, a card level check is made to ensure that the card protection is still in place. If the check passes, the card is tested to see if both the active and standby cards are up or both are down. If the active card is up and the standby card is down, the cardpair fails REX testing.

- POSTCHECK

  When all card testing has completed successfully, shelf level post checks are invoked to ensure the shelf is still in a sane state. REX then closes the TraceLog, sends a REX successful message alarm with the equipment tested, sets its state to IDLE, and sets a wake up for the next START_TIME.

Prerequisites

- Ensure REX has been provisioned according to REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions (page 19).
- You must have the supported cards in the supported configurations on the shelf with no ATM.

Procedure steps to check the current state of REX on the switch

Step	Action
1	Log in to the Media Gateway 15000.
2	Issue the following command: **d rex currentState** The response indicates the current REX test activity on the switch as one of the following: UNINITIALIZED IDLE STOPPING PRECHECK TESTING_CARD<n> POSTCHECK

--End--

Accessing the traceLog files produced from REX executions

You can access the traceLog files from the REX executions using FTP. The REX tracelog file is stored on the CP disk. To retrieve the ASCII log file you must FTP to the CP, with the required access permission, and access the appropriate REX log directory. REX log files for automated and manual executions are stored separately on the CP disk under the following directories:

- For scheduled REX executions:

  ```
  /system/rex/logs/auto/<rex.auto file_name.log>
  ```
- For manual REX executions:

  ```
  /system/rex/logs/manual/<rex.manual file_name.log>
  ```

One log file is generated for each scheduled or manual REX execution. For scheduled REX executions, seven log files are kept, one for each day of the week. For manual REX execution the most recent five log files are kept. When the maximum allowed number of logs per REX execution type is reached, the oldest log file is deleted to allow for the creation of a new log file.

Prerequisites

- Ensure REX has been provisioned according to REX configuration and scheduling for the PT-IP GE and UA-IP GE solutions (page 19).
- Ensure that scheduled automated REX testing is not in progress.
- Ensure you have root user access.

Procedure steps to access the traceLog files

Step	Action
1	FTP to the Media Gateway 15000 to get the tracelog files from the following directories: • For Scheduled REX executions: /system/rex/logs/auto/<rex.auto file_name.log> • For Manual REX executions: /system/rex/logs/manual/<rex.manual file_name.log>

--End--

Interpreting REX results

This section describes the results that are available from REX. Your careful analysis of the alarms and logs generated from REX testing allows you to proactively correct problems that could hinder a card pair's switch over to protection on the MG15000. REX generates logs and alarms for every test execution. The log results indicate pass/fail, time tested, automatic or manual testing, and cards tested. Results are available from the following:

- Message alarms (page 44)
- Alarms (page 45)
- Log files (page 48)

Message alarms

The system issues a message alarm to indicate the start of a REX execution and to inform you of any events or problems encountered throughout the REX execution. Each message alarm provides an appropriate reason for its launch whenever REX testing returns failures. These message alarms include text that is informative as well as timely covering such instances as:

- unsafe to execute Routine Exercise Test for this hardware pair
- test failed for this field replaceable unit
- failed to return to load-sharing operation for this hardware pair

A message alarm also indicates the successful completion of REX testing and contains a list of the hardware that passed REX execution.

Alarms

All alarms are described in detail in *Nortel Multiservice Switch 6400/7400/ 15000/20000 Alarms Reference* (NN10600-500).

A check is made of active alarms before and after REX has run. Alarms that existed before REX testing may be regenerated with a new timestamp after REX testing has occurred because the REX test may detect the same alarm condition. For example, a card pair with both protecting ports down will re-issue the ports down alarm after REX has run on the card pair. REX compares the active alarms before and after the REX execution to determine if any additional alarms were introduced during the REX run. If additional alarms are detected in the shelf postchecks, the REX application reports a failure due to postchecks failing.

The following REX alarms are attributable to the REX execution and are raised when REX starts a run, when REX is disabled, or when REX aborts due to a failure:

- 7082 0001 minor/cleared
  - status = set when REX tests start
  - status = clear when REX test completes successfully, aborts because of failure, or is manually suspended
  - contains list of hardware that passed REX tests
  - cleared after REX terminates for current execution cycle
- 7082 0002 minor
  - status = set when REX component is disabled
  - status = clear when REX component is enabled (PROV> set shelf rex enable yes)
- 7082
- 7082 0003 major: This alarm is issued when REX tests abort because of failure. The cause of the failure can include
  - REX tests on a spared card unit failed.
  - Pre-check for REX tests failed.
  - REX tests on a spared card unit failed.
  - Failed to return to load-sharing operation for this card pair.
  - Post-check for REX tests failed.
  - REX is aborted by manual command.
  - Automated test re-scheduled for the next day as manual test is in progress.

  - — REX testing stopped due to provisioned stop or default manual stop timer expiry

It is normal for alarms raised on REX testing to clear at the end of the REX interval. For new alarms and alarms that persist after REX has run, see: *Nortel Multiservice Switch 6400/7400/15000/20000 Alarms Reference* (NN10600-500). If problems persist, refer to *Nortel Multiservice Switch 7400/15000/20000 Troubleshooting* (NN10600-520) for test and repair of faulty components. Alarms associated with MG15000 REX are visible to the OSS in an SCC2 log stream. The following alarms can be expected to raise and clear throughout the duration of a REX execution on the shelf, fabrics, CPs and FPs:

- 70820001 - SET minor processing operationalCondition
- 70120200 - SET critical processing underlyingResourceUnavail
- 70120100 - SET critical equipment processorProblem
- 70115274 - SET minor communications degradedSignal
- 70115273 - SET minor communications degradedSignal
- 70081019 - SET minor equipment processorProblem
- 00001000 - SET critical operator operationalCondition
- 70030001 - SET major qualityOfService thresholdCrossed
- 70020002 - SET major equipment equipmentFailure
- 00001001 - SET critical equipment processorProblem
- 70120101 - MSG indeterminate equipment equipmentFailure
- 70820001 - CLR cleared processing operationalCondition
- 70120100 - CLR cleared equipment processorProblem
- 00000000 - CLR cleared processing underlyingResourceUnavail
- 70115274 - CLR cleared communications degradedSignal
- 70115273 - CLR cleared communications degradedSignal
- 70081019 - CLR cleared equipment processorProblem
- 70030001 - CLR cleared qualityOfService thresholdCrossed
- 70020002 - CLR cleared equipment equipmentFailure

The existence of the following alarms results in the REX shelf-level checks failing which causes REX to terminate:

- 0000 index group with the following sub indexes
  - — 3000 Memory/processor messaging congestion
- 7000 index group range: 0001-0041 subindex

  - Switch Boot Alarm
  - File System Alarm
  - Patch in progress
- 7002 index group range: 0000-1000 subindex
  - BCS alarms
- 7008 index range 1001 1021 subindex
  - File system alarms (File system full, out of sync, etc.)
- 7011 2002 - FP txClock not synchronized to CP3 Clock
- 7011 5002 - Alarm Status for TDM channels
- 7012 0200 - LP not running - resource not available (no card)
- 7012 index group range: 0300-0301subindex
  - standby card encountered resource issue
- 7013 & 7014 index group range: 0001-0011
  - FP low memory alarm
- 7015 network time (XNTP) Synchronization Alarms
- 7017 Network Synchronization
- 7021 index group range: 0001-0014 subindex
  - IP VR memory exhaustion
- 7021 1022 IP VR Static HeartBeat Failure Alarm
- 7026 1002 HW error or SW configuration error on Card
- 7026 index group range- oamEnet failures 3000-3006 subindex

Log files

The system generates log files for every event that occurs during a REX execution. The files are stored in ASCII format on the CP disk and are found in the following directories:

- For scheduled REX executions:

```
/system/rex/logs/auto/<rex.auto file_name.log>
```

- For manual REX executions:

```
/system/rex/logs/manual/<rex.manual file_name.log>
```

One log file is generated for each scheduled or manual REX execution. For scheduled REX executions, seven log files are kept, one for each day of the week. For manual REX execution the most recent five log files are kept. When the maximum allowed number of logs per REX execution type is reached, the oldest log file is deleted to allow for the creation of a new log file.

When REX runs to completion, a detailed log is produced that itemizes the results for every check and test performed. The REX tracelog file is structured to provide the following:

- date
- start time
- the type of check made
- the CAS commands executed
- result of check, PASSED or FAILED
- a failure reason if check failed and an overall result
- stop time

The trace log file is stored on the CP's disk, under directory /spooled/closed (opened)/appl/rex/<file name>.

See the Example of the REX Tracelog file (page 49).

Example of the REX Tracelog file

The following is an example of a REX tracelog file from a REX execution.

```
rex.auto.20050623T121800.003.log
****************************************************
      REX LOG ( automatic execution)
      Date:  23:06:2005
      Start Time:12:18:01
      Solution Type: UA_IPGe
      Node Name: DESIGN_3
      Node Id: 3
****************************************************

*************************
   Config Check
   Start Time:  12:18:01
*************************

   Software download not in Progress ( passed )

   Software upgrade  not in Progress ( passed )

*************************
   Config Check
   Stop Time:  12:18:01
   Overall Result: Passed
*************************

*************************
   AAL PreCheck
   Start Time:  12:18:01
*************************

*************************
   AAL  PreCheck
   Stop Time:  12:18:01
   Overall Result: Passed
*************************

*************************
   Shelf PreCheck
   Start Time:  12:18:01
*************************


1) Disk Checks

   command: d fs syncStatus
   result: Passed (  = synchronized  )
   command: d fs syncProgress
   result: Passed ( = 100 )
   command: d fs usage
   result: Passed (  < 80  )

2) Scs Checks
   command: d sw dld status
   result: Passed ( = inactive )
   command: d sw tidyStatus
   result: Passed ( = inactive )

3) Fabric Checks

   command: d sh fabric/x adminState
   result: Passed ( unlocked )
   command: d sh fabric/x operationalState
   result: Passed ( enabled )
   command: d sh fabric/x availabilityStatus
   result: Passed ( empty )
   command: d sh fabric/x alarmStatus
   result: Passed ( empty )
   command: d sh fabric/y adminState
   result: Passed ( unlocked )
   command: d sh fabric/y operationalState
   result: Passed ( enabled )
   command: d sh fabric/y availabilityStatus
   result: Passed ( empty )
   command: d sh fabric/y alarmStatus
   result: Passed ( empty )


4) Prov Checks

   command: d pr provisioningActivity
   result: Passed (  = none  )
   command: d pr standbyCpActivity
   result: Passed (  = none  )
   command: d pr activityProgress
   result: Passed (  = n/a  )
   command: d pr standbyCpActivityProgress
   result: Passed (  = n/a  )
   command: d pr editViewAddedComponents
   result: Passed ( = 0 )
   command: d pr editViewDeletedComponents
   result: Passed ( = 0 )
   command: d pr editViewChangedComponents
   result: Passed ( = 0 )


5) Card and Lp Checks

   command: d -notab sh ca/* adminState
   result: Passed ( unlocked )
   command: d -notab sh ca/* fabric/x adminState
   result: Passed ( unlocked )
   command: d -notab sh ca/* fabric/x operationalState
   result: Passed ( enabled )
   command: d -notab sh ca/* fabric/y adminState
   result: Passed ( unlocked )
   command: d -notab sh ca/* fabric/y operationalState
   result: Passed ( enabled )
   command: d -notab lp/* adminState
   result: Passed ( unlocked )
   command: d -notab lp/* operationalState
   result: Passed ( enabled )
   command: d -notab lp/* availabilityStatus
   result: Passed ( empty )
   command: d -notab lp/* alarmStatus
   result: Passed ( empty )
```

6) Sparing Checks

```
   command: d -notab sh Card/* SparedServices adminState
   result: Passed ( unlocked )
   command: d -notab sh Card/* SparedServices
operationalState
   result: Passed ( enabled )
   command: d -notab sh Card/* SparedServices
availabilityStatus
   result: Passed ( empty )
   command: d -notab sh Card/* SparedServices
standbyStatus
   result: Passed (  != coldStandby, notSet )
```

7) CP EP Checks

```
   command: d -p sh cpEquipmentProtection
   result: Passed (  = hot  )
```

8) IP Routing Checks

```
   command: d -notab vr/* adminState
   result: Passed ( unlocked )
   command: d -notab vr/* operationalState
   result: Passed ( enabled )
   command: d -notab vr/* usageState
   result: Passed ( active )
   command: d -notab vr/* Pp/* adminState
   result: Passed ( unlocked )
   command: d -notab vr/* Pp/* operationalState
   result: Passed ( enabled )
   command: d -notab vr/* Pp/* usageState
   result: Passed ( active )
   command: d -notab vr/* Pp/* ipp adminState
   result: Passed ( unlocked )
   command: d -notab vr/* Pp/* ipp operationalState
   result: Passed ( enabled )
   command: d -notab vr/* Pp/* ipp usageState
   result: Passed ( active )
   command: d -notab vr/* Ip adminState
   result: Passed ( unlocked )
   command: d -notab vr/* Ip operationalState
   result: Passed ( enabled )
   command: d -notab vr/* Ip usageState
   result: Passed ( active )
   command: d -notab vr/* Ip ospf adminState
   result: Passed ( unlocked )
   command: d -notab vr/* Ip ospf operationalState
   result: Passed ( enabled )
   command: d -notab vr/* Ip ospf usageState
   result: Passed ( active )
   command: d -notab vr/* Ip ospf Nbr/* state
   result: Passed (  != down, attempt, init, twoWay,
exchangeStart, exchange, loading )
   command: d -notab vr/* Ip static routeEntry/*
operationalState
   result: Passed ( enabled )
  command: d -notab vr/* Ip static routeEntry/(protected
= yes) protectionLevel
   result: Passed (  != portProtected, unprotected,
notApplicable )
   command: d -notab vr/* Ip static route/* nexthop/*
operationalState
   result: Passed ( enabled )
```

9) Laps Checks

```
   command: d -notab laps/* adminState
   result: Passed ( unlocked )
   command: d -notab laps/* operationalState
   result: Passed ( enabled )
   command: d -notab laps/* availabilityStatus
   result: Passed ( empty )
   command: d -notab laps/* alarmStatus
   result: Passed ( empty )
   command: d -notab laps/* nearEndRequest
   result: Passed (  != lockoutOfProtection,
forcedSwitch, signalFail, signalDegrade, waitToRestore,
manualSwitch )
   command: d -notab laps/* sts/* adminState
   result: Passed ( unlocked )
   command: d -notab laps/* sts/* lop
   result: Passed (  != on )
   command: d -notab laps/* sts/* ais
   result: Passed (  != on )
   command: d -notab laps/* sts/* rfi
   result: Passed (  != on )
   command: d -notab laps/* sts/* slm
   result: Passed (  != on )
   command: d -notab laps/* sts/* txa
   result: Passed (  != on )
   command: d -notab laps/* sts/* txrdi
   result: Passed (  != on )


10) Network Sync Checks

   command: d networksync adminState
   result: Passed ( unlocked )
   command: d networksync operationalState
   result: Passed ( enabled )
   command: d networksync clockSyncState
   result: Passed (  = synchronized  )


11) Oam Enet Checks

   command: d Lp/0 Oamenet/0 activeStatus
   result: Passed (  = available  )
   command: d Lp/0 Oamenet/0 standbyStatus
   result: Passed (  = available  )


12) UA_IP Shelf Checks

   command: d -notab lp/* Ethernet/* adminState
   result: Passed ( unlocked )
   command: d -notab lp/* Ethernet/* operationalState
   result: Passed ( enabled )
   command: d -notab lp/* Ethernet/* alarmStatus
   result: Passed ( empty )
   command: d -notab lp/* sonet/* adminState
   result: Passed ( unlocked )
   command: d -notab lp/* sonet/* operationalState
   result: Passed ( enabled )
   command: d -notab lp/* sonet/* alarmStatus
   result: Passed ( empty )
   command: d -notab la/* adminState
   result: Passed ( unlocked )
   command: d -notab la/* operationalState
   result: Passed ( enabled )
   command: d -notab Dlep/* adminState
   result: Passed ( unlocked )
   command: d -notab Dlep/* operationalState
   result: Passed ( enabled )
   command: d -notab Dlep/* alarmStatus
   result: Passed ( empty )

13) UA_IP PDR VR Ospf Nbr Checks
   command: d -notab Vr/VOIP Ip Ospf Nbr/10.48.0.97
state
   result: Passed ( = full )
   command: d -notab Vr/VOIP Ip Ospf Nbr/10.48.0.129
state
   result: Passed ( Component not provisioned )

14) UA_IP PDR VR Checks

   command: d -notab Vr/VOIP ip static adminState
   result: Passed ( unlocked )
   command: d -notab Vr/VOIP ip static operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP ip static usageState
   result: Passed ( active )
   command: d -notab Vr/VOIP ip static
heartbeatDeadInterval
   result: Passed (  > 3  )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 heartbeat
   result: Passed (  = enabled  )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 adminState
   result: Passed ( unlocked )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 protectionLevel
   result: Passed (  = cardProtected  )
   command: d -notab Vr/VOIP ip ospf spareInstance
   result: Passed (  = enable  )
   command: d -notab Vr/VOIP ip ospf ecmpStatus
   result: Passed (  = disabled  )
   command: d -notab Vr/VOIP ip ospf nbr/* state
   result: Passed (  != down, attempt, init, twoWay,
exchange, loading )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* usageState
   result: Passed (  = active  )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* heartbeatState
   result: Passed (  = up  )

*************************
   Ping Checks
   Start Time:  12:18:14
*************************
ping -ip(10.15.0.6) Nsta/12 Vgs  Ctrl/mg
Nsta/12 Vgs Ctrl/mediaGateway
   Ping: Received reply of 1472 bytes from 10.15.0.6:
icmpSeq=0, time=0ms
   Ping: Received reply of 1472 bytes from 10.15.0.6:
icmpSeq=1, time=0ms
ok             2005-06-23 12:18:16.11


*************************
   Ping Checks
   Stop Time:  12:18:16
   Overall Result: Passed
*************************

*************************
   Shelf PreCheck
   Stop Time:  12:18:16
   Overall Result: Passed
*************************

*************************
   Card PreCheck (card 13)
   Start Time:  12:18:16
*************************


1) VSP Card Spare LP Checks

   command: d -notab Lp/13 adminState
   result: Passed ( unlocked )
   command: d -notab Lp/13 operationalState
   result: Passed ( enabled )
   command: d -notab Lp/13 alarmStatus
   result: Passed ( empty )
   command: d -notab Lp/13 cpuUtil
   result: Passed (  < 80  )


2) VSP Card Dlep VSP Checks

   command: d -notab Dlep/6 Vsp adminState
   result: Passed ( unlocked )
   command: d -notab Dlep/6 Vsp operationalState
   result: Passed ( enabled )

*************************
   Card PreCheck (card 13)
   Stop Time:  12:18:18
   Overall Result: Passed
*************************

*************************
   Card Reset Test (card 13)
   Start Time:  12:18:18
*************************
  Received Card Down
  Received Card Up
  Received Lp Up

*************************
   Card Reset Test (card 13)
   Stop Time:  12:27:02
   Overall Result: Passed
*************************

*************************
   Card PostCheck (card 13)
   Start Time:  12:27:02
*************************


1) VSP Card Spare LP Checks

   command: d -notab Lp/13 adminState
   result: Passed ( unlocked )
   command: d -notab Lp/13 operationalState
   result: Passed ( enabled )
   command: d -notab Lp/13 alarmStatus
   result: Passed ( empty )
   command: d -notab Lp/13 cpuUtil
   result: Passed (  < 80  )


2) VSP Card Dlep VSP Checks

   command: d -notab Dlep/6 Vsp adminState
   result: Passed ( unlocked )
   command: d -notab Dlep/6 Vsp operationalState
   result: Passed ( enabled )

*************************
   Card PostCheck (card 13)
   Stop Time:  12:27:04
   Overall Result: Passed
*************************

*************************
   Card PreCheck (card 1)
   Start Time:  12:27:04
*************************


1) CP3 Card Checks

   command: d fs usage
   result: Passed (  < 80  )
   command: d fs syncStatus
   result: Passed (  = synchronized  )
   command: d fs syncProgress
   result: Passed ( = 100 )
   command: d sh ca/0 cpuUtil
   result: Passed (  < 80  )
   command: d sh ca/1 cpuUtil
   result: Passed (  < 80  )
   command: d Lp/0 Oamenet/0 activeStatus
   result: Passed (  = available  )
   command: d Lp/0 Oamenet/0 standbyStatus
   result: Passed (  = available  )

*************************
   Card PreCheck (card 1)
   Stop Time:  12:27:05
   Overall Result: Passed
*************************

*************************
   Card Reset Test (card 1)
   Start Time:  12:27:05
*************************
  Received Card Down
  Received Card Up
  Received Lp Up

*************************
   Card Reset Test (card 1)
   Stop Time:  12:33:38
   Overall Result: Passed
*************************

*************************
   Card PostCheck (card 1)
   Start Time:  12:33:38
*************************
```

```
1) CP3 Card Checks

   command: d fs usage
   result: Passed (  < 80  )
   command: d fs syncStatus
   result: Passed (  = synchronized  )
   command: d fs syncProgress
   result: Passed ( = 100 )
   command: d sh ca/0 cpuUtil
   result: Passed (  < 80  )
   command: d sh ca/1 cpuUtil
   result: Passed (  < 80  )
   command: d Lp/0 Oamenet/0 activeStatus
   result: Passed (  = available  )
   command: d Lp/0 Oamenet/0 standbyStatus
   result: Passed (  = available  )

*************************
   Card PostCheck (card 1)
   Stop Time:  12:33:39
   Overall Result: Passed
*************************

*************************
   Card PreCheck (card 14)
   Start Time:  12:33:40
*************************


1) GIGE Card Checks

   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 heartbeat
   result: Passed (  = enabled  )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 adminState
   result: Passed ( unlocked )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 protectionLevel
   result: Passed (  = cardProtected  )
   command: d -notab Vr/VOIP ip ospf nbr/* state
   result: Passed (  != down, attempt, init, twoWay,
exchange, loading )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* usageState
   result: Passed (  = active  )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* heartbeatState
   result: Passed (  = up  )

*************************
   Card PreCheck (card 14)
   Stop Time:  12:33:41
   Overall Result: Passed
*************************

*************************
   Card Reset Test (card 14)
   Start Time:  12:33:41
*************************
  Received Card Down
  Received Card Up
  Received Lp Up

*************************
   Card Reset Test (card 14)
   Stop Time:  12:41:39
   Overall Result: Passed
*************************

*************************
   Card PostCheck (card 14)
   Start Time:  12:41:39
*************************


1) GIGE Card Checks

   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 heartbeat
   result: Passed (  = enabled  )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 adminState
   result: Passed ( unlocked )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 protectionLevel
   result: Passed (  = cardProtected  )
   command: d -notab Vr/VOIP ip ospf nbr/* state
   result: Passed (  != down, attempt, init, twoWay,
exchange, loading )
```

```
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* usageState
   result: Passed (  = active  )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* heartbeatState
   result: Passed (  = up  )

*************************
   Card PostCheck (card 14)
   Stop Time:  12:41:40
   Overall Result: Passed
*************************

*************************
   AAL PostCheck
   Start Time:  12:41:40
*************************

*************************
   AAL  PostCheck
   Stop Time:  12:41:40
   Overall Result: Passed
*************************

*************************
   Shelf PostCheck
   Start Time:  12:41:40
*************************


1) Disk Checks

   command: d fs syncStatus
   result: Passed (  = synchronized  )
   command: d fs syncProgress
   result: Passed ( = 100 )
   command: d fs usage
   result: Passed (  < 80  )


2) Fabric Checks

   command: d sh fabric/x adminState
   result: Passed ( unlocked )
   command: d sh fabric/x operationalState
   result: Passed ( enabled )
   command: d sh fabric/x availabilityStatus
   result: Passed ( empty )
   command: d sh fabric/x alarmStatus
   result: Passed ( empty )
   command: d sh fabric/y adminState
   result: Passed ( unlocked )
   command: d sh fabric/y operationalState
   result: Passed ( enabled )
   command: d sh fabric/y availabilityStatus
   result: Passed ( empty )
   command: d sh fabric/y alarmStatus
   result: Passed ( empty )


3) Prov Checks

   command: d pr provisioningActivity
   result: Passed (  = none  )
   command: d pr standbyCpActivity
   result: Passed (  = none  )
   command: d pr activityProgress
   result: Passed (  = n/a  )
   command: d pr standbyCpActivityProgress
   result: Passed (  = n/a  )
   command: d pr editViewAddedComponents
   result: Passed ( = 0 )
   command: d pr editViewDeletedComponents
   result: Passed ( = 0 )
   command: d pr editViewChangedComponents
   result: Passed ( = 0 )


4) Card and Lp Checks

   command: d -notab sh ca/* adminState
   result: Passed ( unlocked )
   command: d -notab sh ca/* fabric/x adminState
   result: Passed ( unlocked )
   command: d -notab sh ca/* fabric/x operationalState
   result: Passed ( enabled )
   command: d -notab sh ca/* fabric/y adminState
   result: Passed ( unlocked )
   command: d -notab sh ca/* fabric/y operationalState
   result: Passed ( enabled )
   command: d -notab lp/* adminState
   result: Passed ( unlocked )
   command: d -notab lp/* operationalState
   result: Passed ( enabled )
   command: d -notab lp/* availabilityStatus
   result: Passed ( empty )
   command: d -notab lp/* alarmStatus
   result: Passed ( empty )
```

```
5) Sparing Checks

   command: d -notab sh Card/* SparedServices adminState
   result: Passed ( unlocked )
   command: d -notab sh Card/* SparedServices
operationalState
   result: Passed ( enabled )
   command: d -notab sh Card/* SparedServices
availabilityStatus
   result: Passed ( empty )
   command: d -notab sh Card/* SparedServices
standbyStatus
   result: Passed (  != coldStandby, notSet )


6) CP EP Checks

   command: d -p sh cpEquipmentProtection
   result: Passed (  = hot  )


7) IP Routing Checks

   command: d -notab vr/* adminState
   result: Passed ( unlocked )
   command: d -notab vr/* operationalState
   result: Passed ( enabled )
   command: d -notab vr/* usageState
   result: Passed ( active )
   command: d -notab vr/* Pp/* adminState
   result: Passed ( unlocked )
   command: d -notab vr/* Pp/* operationalState
   result: Passed ( enabled )
   command: d -notab vr/* Pp/* usageState
   result: Passed ( active )
   command: d -notab vr/* Pp/* ipp adminState
   result: Passed ( unlocked )
   command: d -notab vr/* Pp/* ipp operationalState
   result: Passed ( enabled )
   command: d -notab vr/* Pp/* ipp usageState
   result: Passed ( active )
   command: d -notab vr/* Ip adminState
   result: Passed ( unlocked )
   command: d -notab vr/* Ip operationalState
   result: Passed ( enabled )
   command: d -notab vr/* Ip usageState
   result: Passed ( active )
   command: d -notab vr/* Ip ospf adminState
   result: Passed ( unlocked )
   command: d -notab vr/* Ip ospf operationalState
   result: Passed ( enabled )
   command: d -notab vr/* Ip ospf usageState
   result: Passed ( active )
   command: d -notab vr/* Ip ospf Nbr/* state
   result: Passed (  != down, attempt, init, twoWay,
exchangeStart, exchange, loading )
   command: d -notab vr/* Ip static routeEntry/*
operationalState
   result: Passed ( enabled )
  command: d -notab vr/* Ip static routeEntry/(protected
= yes) protectionLevel
   result: Passed (  != portProtected, unprotected,
notApplicable )
   command: d -notab vr/* Ip static route/* nexthop/*
operationalState
   result: Passed ( enabled )
```

8) Laps Checks

```
   command: d -notab laps/* adminState
   result: Passed ( unlocked )
   command: d -notab laps/* operationalState
   result: Passed ( enabled )
   command: d -notab laps/* availabilityStatus
   result: Passed ( empty )
   command: d -notab laps/* alarmStatus
   result: Passed ( empty )
   command: d -notab laps/* nearEndRequest
   result: Passed (  != lockoutOfProtection,
forcedSwitch, signalFail, signalDegrade, waitToRestore,
manualSwitch )
   command: d -notab laps/* sts/* adminState
   result: Passed ( unlocked )
   command: d -notab laps/* sts/* lop
   result: Passed (  != on )
   command: d -notab laps/* sts/* ais
   result: Passed (  != on )
   command: d -notab laps/* sts/* rfi
   result: Passed (  != on )
   command: d -notab laps/* sts/* slm
   result: Passed (  != on )
   command: d -notab laps/* sts/* txa
   result: Passed (  != on )
   command: d -notab laps/* sts/* txrdi
   result: Passed (  != on )
```

9) Network Sync Checks

```
   command: d networksync adminState
   result: Passed ( unlocked )
   command: d networksync operationalState
   result: Passed ( enabled )
   command: d networksync clockSyncState
   result: Passed (  = synchronized  )


10) Oam Enet Checks

   command: d Lp/0 Oamenet/0 activeStatus
   result: Passed (  = available  )
   command: d Lp/0 Oamenet/0 standbyStatus
   result: Passed (  = available  )


11) UA_IP Shelf Checks

   command: d -notab lp/* Ethernet/* adminState
   result: Passed ( unlocked )
   command: d -notab lp/* Ethernet/* operationalState
   result: Passed ( enabled )
   command: d -notab lp/* Ethernet/* alarmStatus
   result: Passed ( empty )
   command: d -notab lp/* sonet/* adminState
   result: Passed ( unlocked )
   command: d -notab lp/* sonet/* operationalState
   result: Passed ( enabled )
   command: d -notab lp/* sonet/* alarmStatus
   result: Passed ( empty )
   command: d -notab la/* adminState
   result: Passed ( unlocked )
   command: d -notab la/* operationalState
   result: Passed ( enabled )
   command: d -notab Dlep/* adminState
   result: Passed ( unlocked )
   command: d -notab Dlep/* operationalState
   result: Passed ( enabled )
   command: d -notab Dlep/* alarmStatus
   result: Passed ( empty )


12) UA_IP PDR VR Checks

   command: d -notab Vr/VOIP ip static adminState
   result: Passed ( unlocked )
   command: d -notab Vr/VOIP ip static operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP ip static usageState
   result: Passed ( active )
   command: d -notab Vr/VOIP ip static
heartbeatDeadInterval
```

```
   result: Passed (  > 3  )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 heartbeat
   result: Passed (  = enabled  )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 adminState
   result: Passed ( unlocked )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP ip static route/
0.0.0.0,0.0.0.0,0 protectionLevel
   result: Passed (  = cardProtected  )
   command: d -notab Vr/VOIP ip ospf spareInstance
   result: Passed (  = enable  )
   command: d -notab Vr/VOIP ip ospf ecmpStatus
   result: Passed (  = disabled  )
   command: d -notab Vr/VOIP ip ospf nbr/* state
   result: Passed (  != down, attempt, init, twoWay,
exchange, loading )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* operationalState
   result: Passed ( enabled )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* usageState
   result: Passed (  = active  )
   command: d -notab Vr/VOIP Ip static route/
0.0.0.0,0.0.0.0,0 nexthop/* heartbeatState
   result: Passed (  = up  )

*************************
   Ping Checks
   Start Time:  12:41:52
*************************
ping -ip(10.15.0.6) Nsta/12 Vgs  Ctrl/mg
Nsta/12 Vgs Ctrl/mediaGateway
   Ping: Received reply of 1472 bytes from 10.15.0.6:
icmpSeq=0, time=0ms
   Ping: Received reply of 1472 bytes from 10.15.0.6:
icmpSeq=1, time=0ms
ok             2005-06-23 12:41:54.94


*************************
   Ping Checks
   Stop Time:  12:41:54
   Overall Result: Passed
*************************

*************************
   Shelf PostCheck
   Stop Time:  12:41:54
   Overall Result: Passed
*************************
****************************************************
   REX Execution Complete
   Stop Time:  12:41:54
   Overall Result:  Passed
****************************************************
```

Supporting information

This section describes the REX system behavior for the checks that are performed when REX is activated and run to completion. The Shelf configuration prechecks, and the card prechecks sections describe the prechecks that REX performs before the actual diagnostic card tests. The postchecks section describes the postchecks that REX performs after the card tests. There are various checks performed at each level to ensure that REX does not introduce any new problems and to detect any problems that could hinder REX from running to completion. For a complete list of REX prechecks and postchecks, see the Example of the REX Tracelog file (page 49).

Shelf configuration checks

Shelf health and configuration prechecks ensure that there are no system wide issues that will worsen as a result of REX application execution. The prechecks at this level ensure certain minimum conditions exist that will allow the REX application to start individual card testing. The shelf level check ensures the MG15000 is in a reasonable system wide state for card level testing to start.

The shelf checks start with the gathering of alarms on the shelf. Then a check is made to ensure that there are only supported cards present on the shelf. If there are cards or configurations that REX does not support, the check will fail and REX will raise alarms and stop. The next test checks to ensure that there are synchronized disks and that the operational state of the disks is enabled with administrative states unlocked. Disk usage should be below 80% to ensure that there is space for the REX trace file to be stored. Then a check is made to ensure that the operational states of the fabrics are enabled. A check of all fabric ports is made to ensure that they are all enabled. A check is then made of the prov component is made to ensure that an upgrade is not currently in progress.

Card checks at this level are physical checks only. All cards must be in an unlocked and enabled state. This avoids having the REX application run during a card swap maintenance procedure. A check of the fabric port status on the card must be enabled. This ensures that the card can deliver full datapath services through the fabric. REX also verifies the card uptime at this level. Any cards in service for less than 24 hours are not eligible for REX testing. A check is then made to ensure that the standby status of the spared card is not in a degraded state. At the port layer, a check of the LAPs availability status is made to ensure that there is no degradation in simplex operations.

At this level, a check of the vr ip static routes is made to ensure the protected state is “yes”. The operational state must be enabled and the administrative state unlocked. The vr ip ospf provisionable attribute *spareInstance* is verified to ensure that it is set to enable and the attribute *ecmpStatus* is set to disabled.

Finally, a series of pings are issued to confirm connectivity with the known Gateway Controllers (GWC).

If you do a manual start of REX without the cardPairsToTest parameter populated, you can perform the shelf prechecks only.

Card prechecks

Card prechecks ensure that there are no local issues that could worsen as a result of REX application execution. Prechecks are performed on the following:

- CP3 cards

  CP3 prechecks involve checking disks to ensure they are synchronized and to ensure there is at least 20% disk space available, a check of memory utilization and CPU utilization is made to ensure that the MG15000 is not experiencing a congestion condition. As well, the oamEnet is examined to ensure there is communication to the MDMs.

- Fabric

  Fabric prechecks involve examining the port status of both fabrics to ensure they are all operationally enabled.

- 4pGigE

  4pGigE prechecks involve checking the card pair status to ensure it is operationally enabled and in-service. The precheck verifies that the card has been in service for at least 24 hours. A check is made of the protected default route state and the OSPF neighbors to ensure they are enabled. All configured ports must be enabled and unlocked.

- VSP3-o

  The VSP3o prechecks consist of verifying that there is an inservice spare card out of a pair, or that they are both down. The alarm status on the TDM channels is determined, as well as, the TDM status and the Laps of the spared ports.

- 2pVSP4e

  The 2pVSP4e prechecks consist of verifying that there is an inservice spare card out of a pair, or that they are both down. The alarm status on the TDM channels is determined, as well as, the TDM status and the Laps of the spared ports.

Postchecks

Post checks ensure that there are no issues introduced as a result of REX execution. The postchecks run after all shelf and card testing are complete and consist of running the precheck criteria to ensure that the system is returned to a normal state.

A check of the active alarm list is performed to ensure that no new alarms are present after REX execution. The spare CP3 is tested to ensure that the spare CP3 has returned to a proper spared condition. The fabric is tested to examine the spare fabric card and ensure that it has returned to service and is operationally enabled. A check of fabric ports is also made to ensure that they are providing service.

If a CP switchover occurs during the fabric test, REX detects that the given fabric is locked at the postcheck level.

Final logs are then written to the traceLog file.

Nortel Media Gateway 7480/15000

Periodic Routine Exercise

Sourced in Canada and the United States of America.

Publication: NN10600-783
Document status: Standard
Document issue: 7.2S3
Document date: October 2006
Product release: PCR 7.2
Job function: Operations
Type: NTP
Language type: U.S. English

To provide feedback or report a problem with this document, go to www.nortel.com/documentfeedback.