Retail passion. Innovative solutions.

Retek® Integration Bus™ 11.1

Diagnostic and Monitoring Tool Kit

Corporate Headquarters:

Retek Inc.
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403
USA

888.61.RETEK (toll free US)
Switchboard:
+1 612 587 5000

Fax:
+1 612 587 5100

European Headquarters:

Retek
110 Wigmore Street
London
W1U 3RW
United Kingdom

Switchboard:
+44 (0)20 7563 4600

Sales Enquiries:
+44 (0)20 7563 46 46

Fax:
+44 (0)20 7563 46 10

Customer Support

Customer Support hours

Customer Support is available 7x24x365 via email, phone, and Web access.

Depending on the Support option chosen by a particular client (Standard, Plus, or Premium), the times that certain services are delivered may be restricted. Severity 1 (Critical) issues are addressed on a 7x24 basis and receive continuous attention until resolved, for all clients on active maintenance. Retek customers on active maintenance agreements may contact a global Customer Support representative in accordance with contract terms in one of the following ways.

Contact Method	Contact Information
E-mail	support@retek.com
Internet (ROCS)	rocs.retek.com Retek's secure client Web site to update and view issues
Phone	+1 612 587 5800

Toll free alternatives are also available in various regions of the world:

Australia	+1 800 555 923 (AU-Telstra) or +1 800 000 562 (AU-Optus)
France	0800 90 91 66
Hong Kong	800 96 4262
Korea	00 308 13 1342
United Kingdom	0800 917 2863
United States	+1 800 61 RETEK or 800 617 3835

Mail

Retek Customer Support
Retek on the Mall
950 Nicollet Mall
Minneapolis, MN 55403

When contacting Customer Support, please provide:

- Product version and program/module name.
- Functional and technical description of the problem (include business impact).
- Detailed step-by-step instructions to recreate.
- Exact error message received.
- Screen shots of each step you take.

Contents

Chapter 1 – Overview

Disclaimer: These scripts are not a supported Retek product. They are provided as a set of utility scripts to Retek RIB customers for reference purposes. No warranties are given or implied.

The purpose of the tool kit is to provide example scripts to address three areas:

- Operations (scanning and monitoring)
- Production debug (scanning and quick triage)
- Test debug (scanning and fine grain control)

There are scripts to monitor the RIB in a production setting and to aid in the diagnosis of RIB issues, as well as individual tools to monitor and manipulate the e*ways and the JMS in test and development setting.

The scripts are written to be stand-a-lone tools that can be integrated into any higher level framework, such as email notifications or SMNP-base frameworks such as Tivoli or HP OpenView.

There is a very simple character-based menu system provided to allow a single point of integration.

Functionality:

- Support for Multiple JMS ports.
- Support Multiple RWMS databases.
- Ability to run the tool kit scripts on a remote host via rsh, so all scripts has been edited to support rsh invocation.
- Rib health script that invokes several of the tools and determines levels of criticality (DOWN, FAIL, and WARNING).
- Support for different email groups (critical and warning)
- Log forwarding to a central log host.
- Submenus to reduce size and complexity of selections.

Chapter 2 – Objectives

There are several specific requirements that were used in developing these tools:

- Requirement 1: The tools must be executable on the RIB/SeeBeyond Host.
- Requirement 2: The tools must be written in standard UNIX shell and UNIX shell tools, and where necessary and applicable; Java/JDBC programs.
- Requirement 3: The tools must be stand-alone and parameter driven so they can be integrated as components in a larger strategy, such as integration with an email notification system, paging system or SNMP Framework such as HP OpenView's ITO.
- Requirement 4: The tools must be designed to be environment configurable, with the configurations being driven by setup files.
- Requirement 5: The tools must be UNIX system agnostic. They must run on any of the platforms support by Retek and SeeBeyond.
- Requirement 6: The tools must be self-contained and require no additional runtime components be installed by the host system administrators, but they expect the RIB and SeeBeyond environment.

Chapter 3 – Installation

The tools are all written for a UNIX system, and are intended to be as Unix-OS agnostic as possible. They are packaged as a UNIX tar file; rdmtXX_eng_ga.tar

The tar file can be expanded to whatever location desired, the tools are all written to run relative to the ~/rdmtXX directory.

They should be installed by the egate user on the system that runs the RIB/SeeBeyond environment. The recommended location is $HOME/rdmtXX, but that is a preference (if not then see configuration below).

Once the tar file has been expanded, add the location to the egate users PATH in the .profile:

```
PATH=:$PATH:$HOME/rdmtXX
```

Because there are configurations that contain passwords, the setup.sh script will set the permissions be to 700 (-rwx------) on all files within the rdmtXX directory structure.

There are a few java programs used by the scripts, they are located in the ~/rdmtXX/java directory.

The installation script setup.sh will make the basic configuration settings need to run the tools in the tool kit.

After the installation, these configurations can be changed at anytime via any text editor to the appropriate config file.

To perform the basic install and configuration, there is a setup.sh script supplied with the toolkit.

- cd :$HOME/rdmtXX
- chmod 700 setup.sh
- ./setup.sh

Configuration parameters

The script will prompt you for several configuration parameters. The script is smart enough to look at the environment of the user executing the script and supply default answers, but be prepared to answer the following questions:

General Environment Parameters	
EHOME	/u00/egate/egate
JMS PORT	24053
REGISTRY PORT	23001
SCHEMA NAME	RIB1102
ADMINSTRATOR PASSWORD	STC
RIBLOGS DIRECTORY	$EHOME/RIBLOGS
RDMTLOGS DIRECTORY	$EHOME/RDMTLOGS

Parameters For Each Hospital (RMS, RDM, SIM, etc…)	
USER NAME	rms
USER PASSWORD	retek
DATABASE HOST	mspdev38
DATABASE LISTENER PORT	1521

Configuration files

The configuration files that drive the behavior of the scripts are located in the ~/rdmtXX directory.

- rdmt.conf
- mfqtables.conf
- uploadtables.conf
- wmaillist.conf
- cmaillist.conf
- pinghosts.conf
- rdmthosts.conf
- loghost.conf

The primary configuration file is the rdmt.conf and the contents are faily self explanatory. The biggest decision to make is the location of the tmp files. Since some of them can get very large, beware of the using the default /tmp directory.

Note: On host systems where there are multiple environments running the tool kit, having them all use the /tmp directory could cause confusion. The recommendation in this situation is to create a tmp directory unique for each environment (i.e. $EHOME/tmp).

Each of the other configuration file are self-explanatory, with examples included.

For the rsh feature to work (remote rdmt) the egate user on both ends need the .rhost file created appropriately. See the man page on rsh and rhosts.

Ping hosts is configured via the pinghosts.conf file. This is intended to be a way to validate the network between the RIB hosts and any other systems that it needs, such as database servers or remote JMS hosts.

Threshold values

There are several user definable thresholds that the scripts look at to determine PASS/FAIL/WARNING conditions. These are located in the rdmt.conf file. These may need to be adjusted for each environment.

```
# Thresholds for triggering alerts
#
hosp_threshold=200                              #  row count for
rib_messages
mfqueue_threshold=100000                #  per table
upload_threshold=100000                   # per table
JMS_topic_threshold=1000                #  per topic
#
# Thresholds for processes virtual memory size -- in pages
#   usually 8192 bytes.
#
stcregd_threshold=12000
stccb_threshold=100000
stcjms_threshold=10000
stceway_threshold=50000
#
```

Chapter 4 – Email support

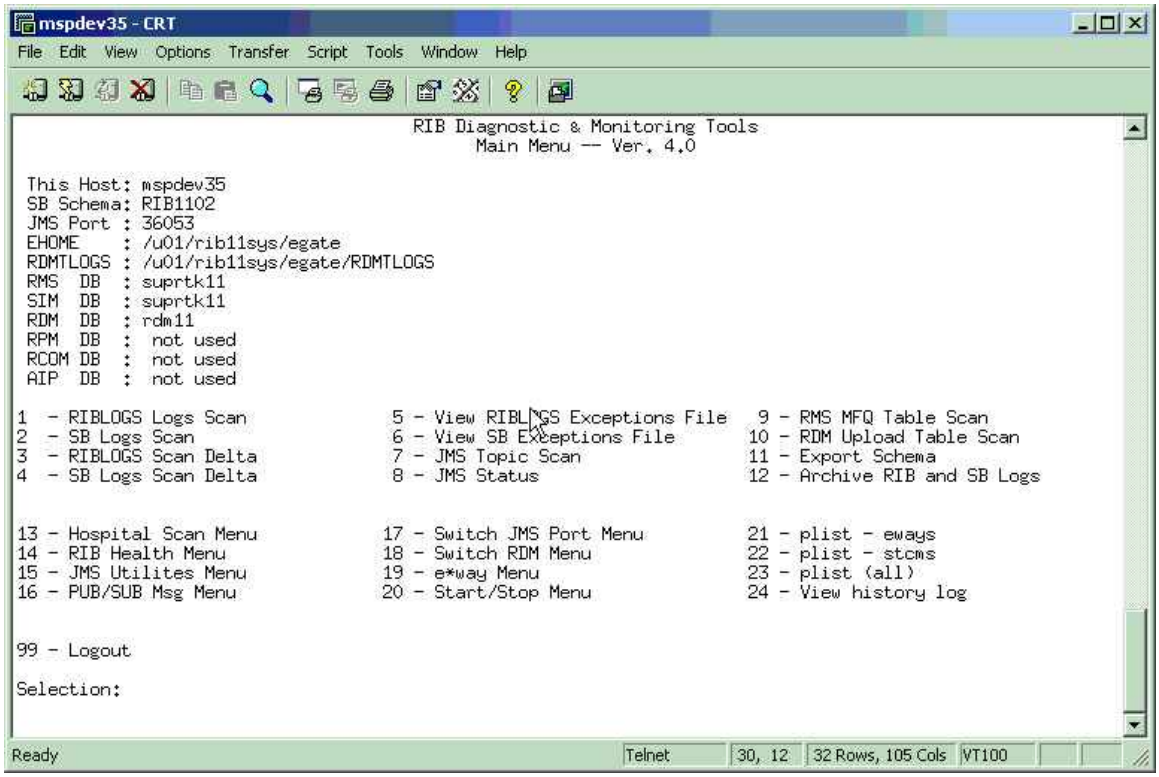
There are several cron jobs that are written to call the basic tool scripts to perform email notifications, as well as necessitated the cloning of some of the scripts to perform silently. These are all suffixed with _email. Several of the scan scripts also have a threshold values in the rdmt.conf file that it uses to determine if email should be sent.

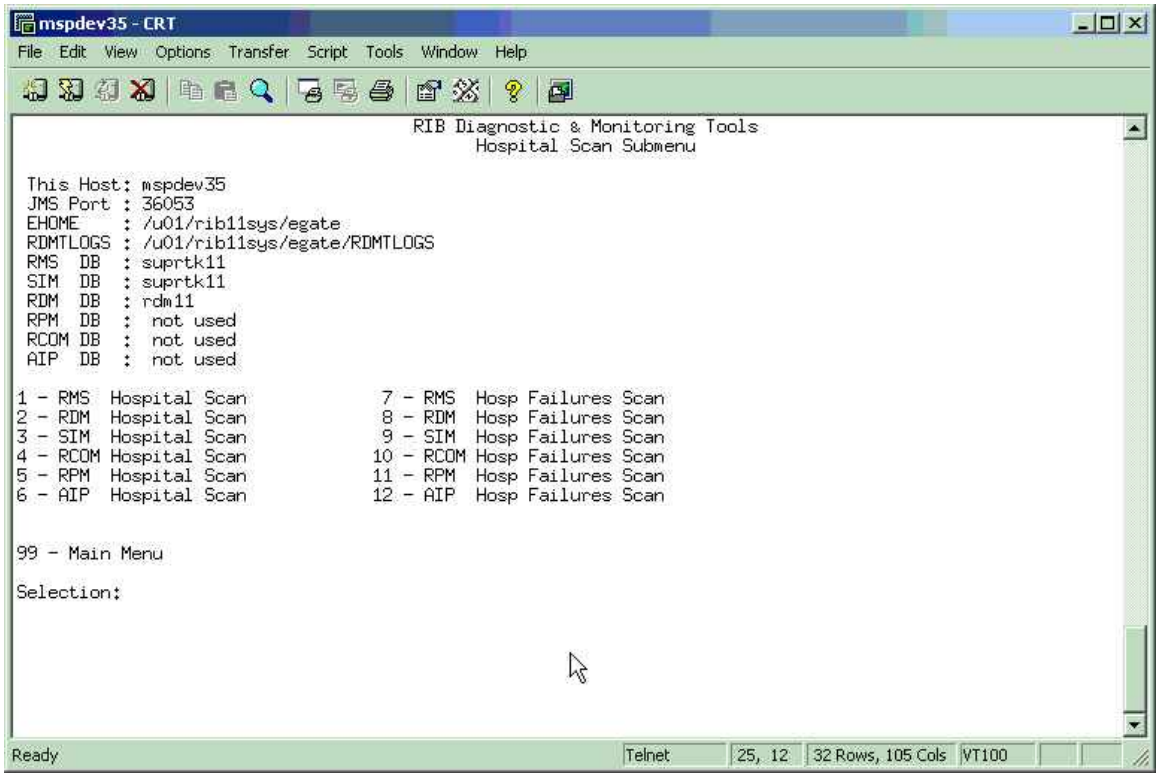
There are two mail configuration files:

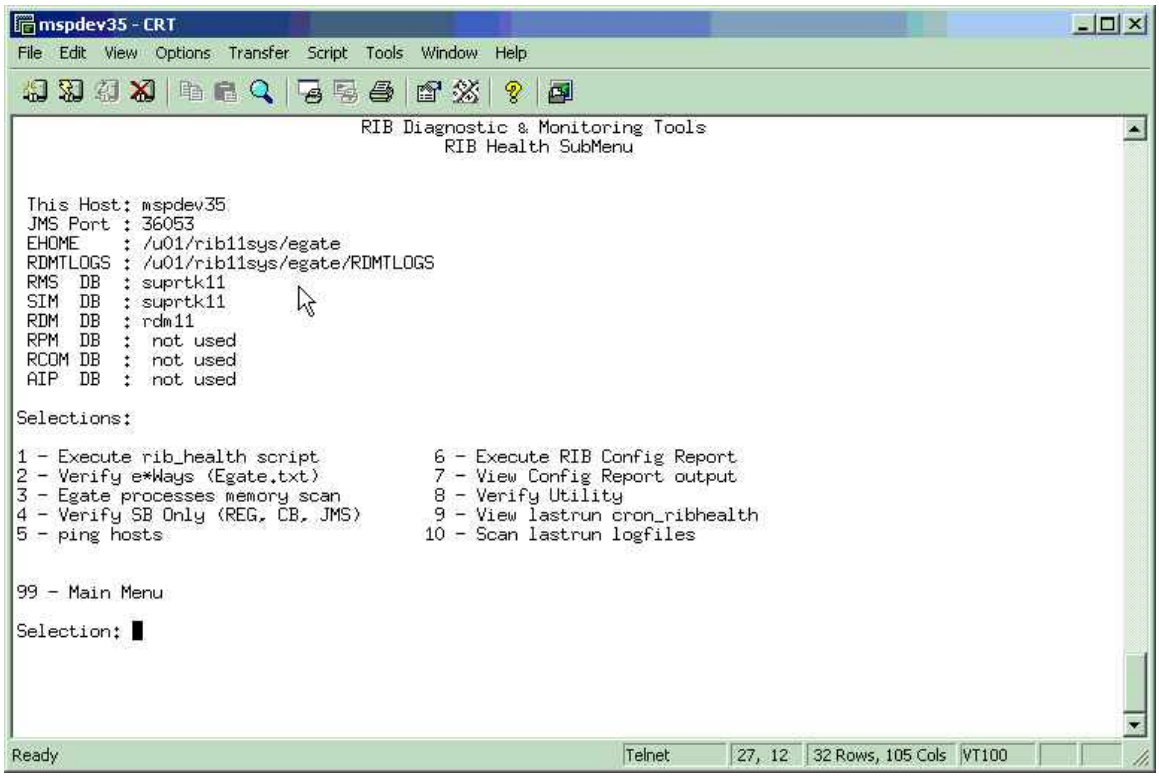
- wmaillist.conf and
- cmaillist.conf.

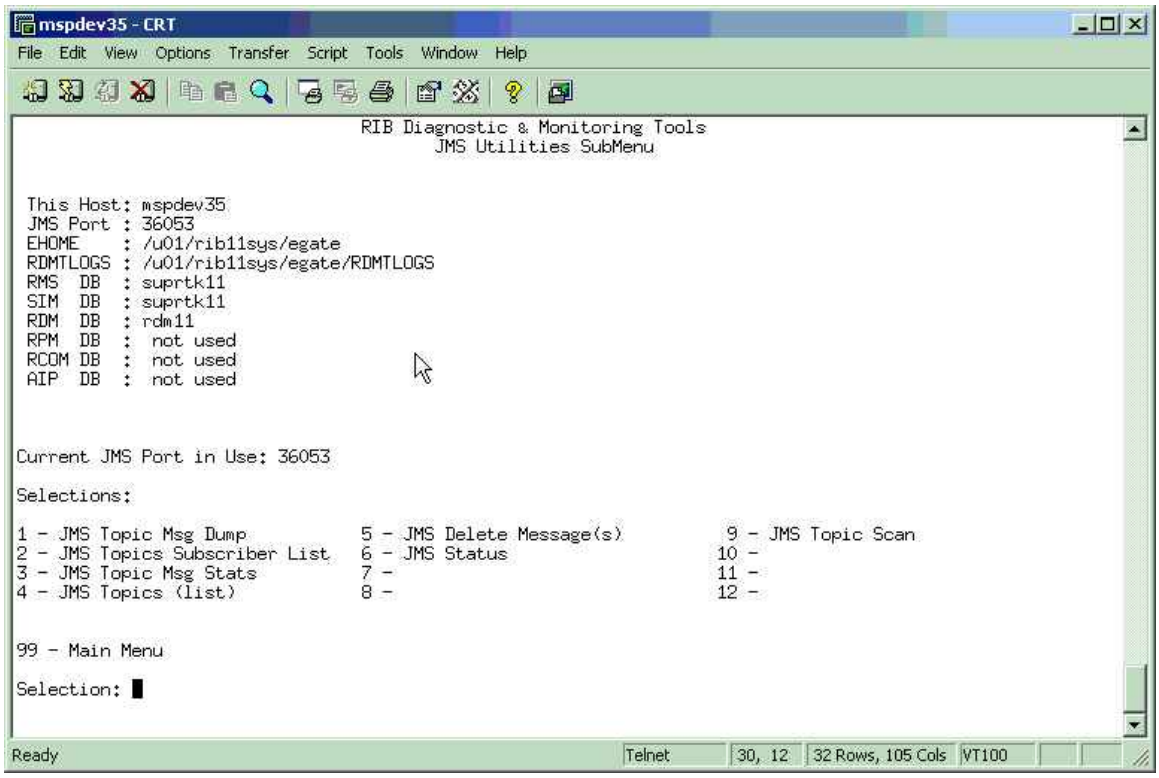
They are for critical and warning mail. The scan scripts produce either UP/DOWN, PASS/FAIL or OK/WARNING. DOWN is considered critical (such as the SeeBeyond processes). FAIL and WARNING are the other group.

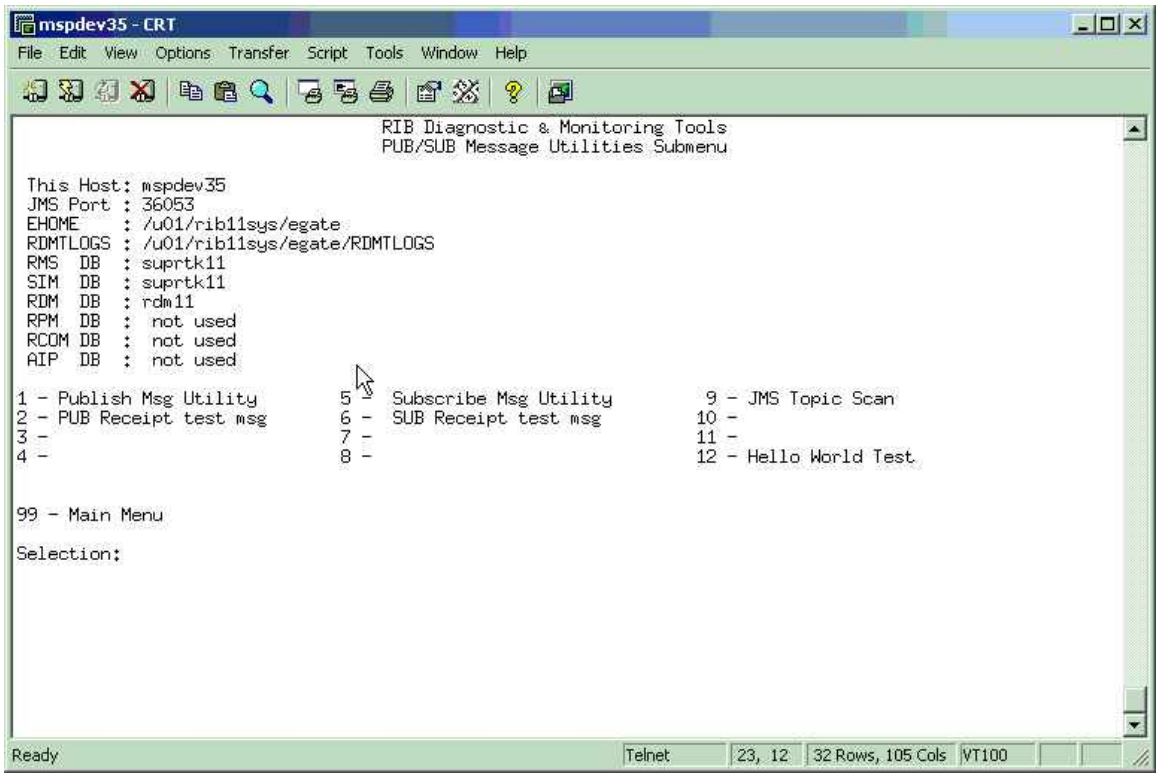
Chapter 5 – RDMT menus

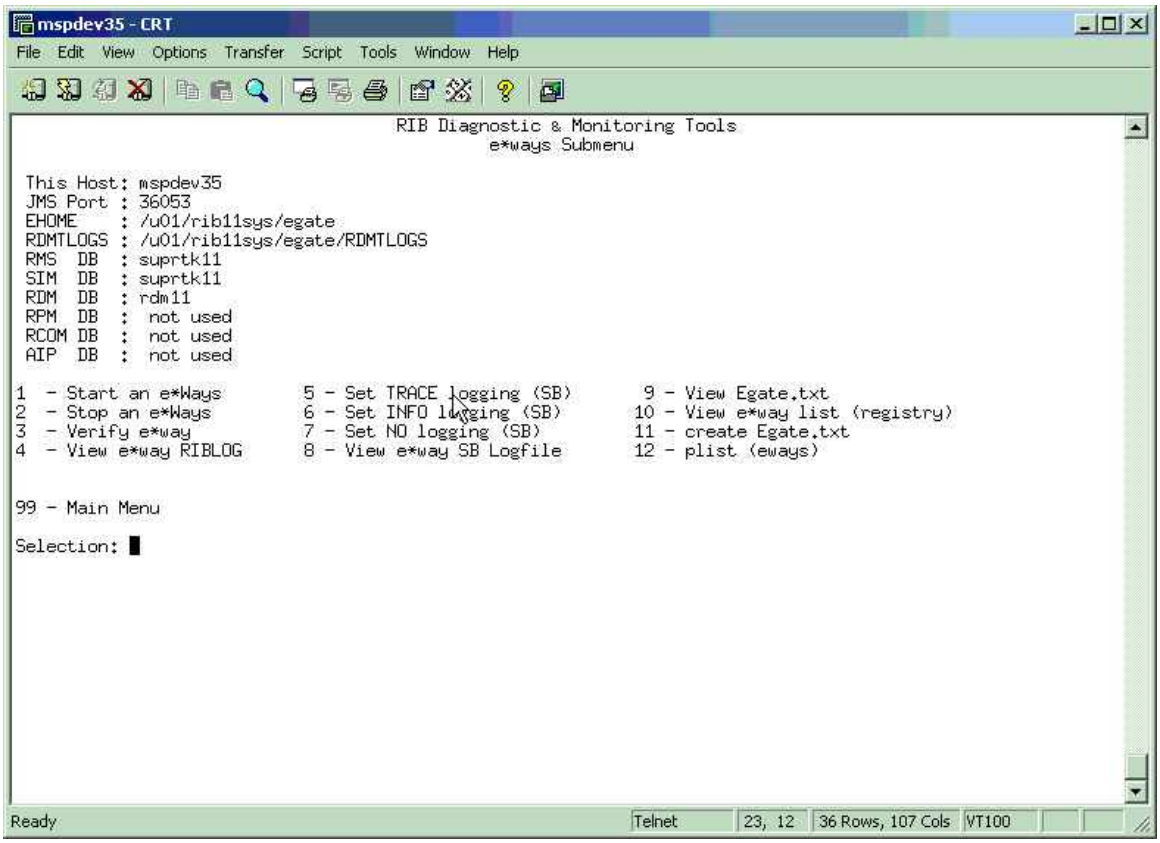


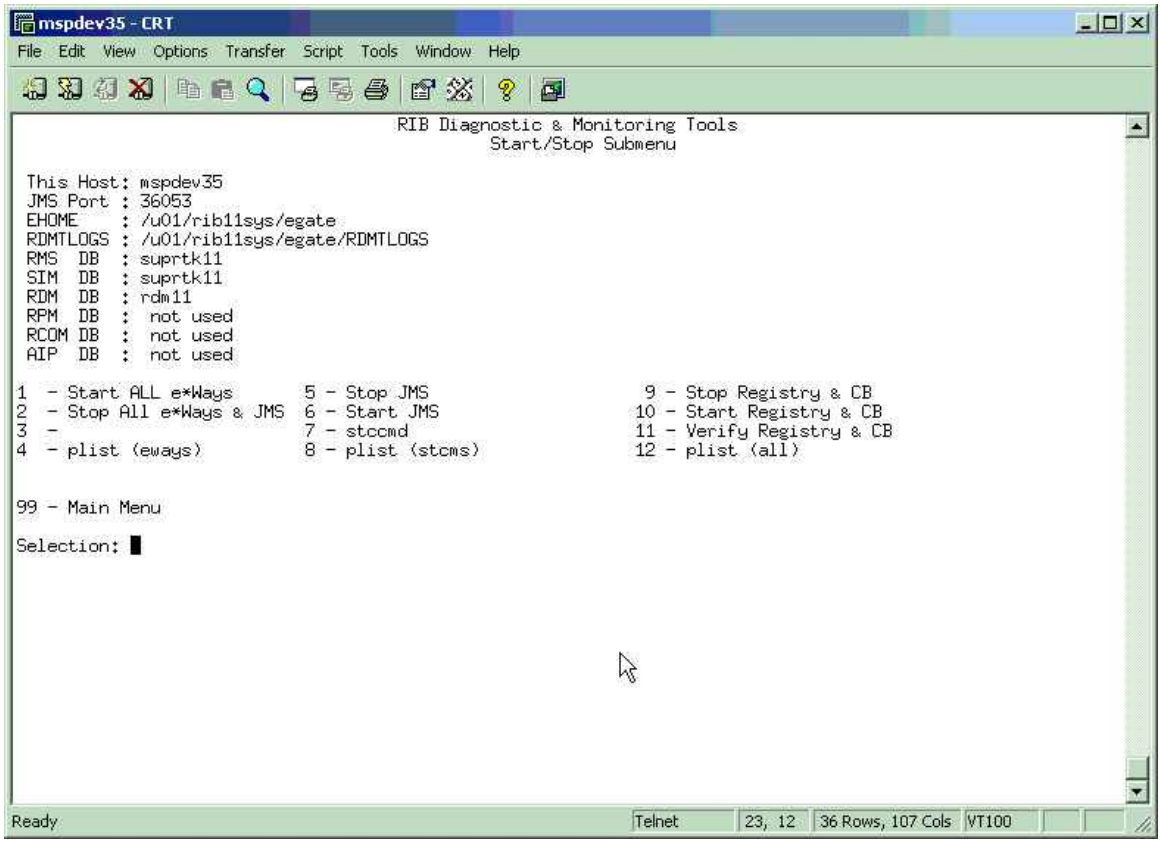



mspdev35 - CRT
File Edit View Options Transfer Script Tools Window Help
RIB Diagnostic & Monitoring Tools
RIB Health SubMenu
This Host: mspdev35
JMS Port : 36053
EHOME : /u01/rib11sys/egate
RDMTLOGS : /u01/rib11sys/egate/RDMTLOGS
RMS DB : suprtk11
SIM DB : suprtk11
RDM DB : rdm11
RPM DB : not used
RCOM DB : not used
AIP DB : not used
Selections:
1 - Execute rib_health script
2 - Verify e*Ways (Egate.txt)
3 - Egate processes memory scan
4 - Verify SB Only (REG, CB, JMS)
5 - ping hosts
6 - Execute RIB Config Report
7 - View Config Report output
8 - Verify Utility
9 - View lastrun cron_ribhealth
10 - Scan lastrun logfiles
99 - Main Menu
Selection:
Ready
Telnet
27, 12
32 Rows, 105 Cols
VT100


mspdev35 - CRT
File Edit View Options Transfer Script Tools Window Help
RIB Diagnostic & Monitoring Tools
JMS Utilities SubMenu
This Host: mspdev35
JMS Port : 36053
EHOME : /u01/rib11sys/egate
RDMTLOGS : /u01/rib11sys/egate/RDMTLOGS
RMS DB : suprtk11
SIM DB : suprtk11
RDM DB : rdm11
RPM DB : not used
RCOM DB : not used
AIP DB : not used
Current JMS Port in Use: 36053
Selections:
1 - JMS Topic Msg Dump
2 - JMS Topics Subscriber List
3 - JMS Topic Msg Stats
4 - JMS Topics (list)
5 - JMS Delete Message(s)
6 - JMS Status
7 -
8 -
9 - JMS Topic Scan
10 -
11 -
12 -
99 - Main Menu
Selection:
Ready
Telnet
30, 12
32 Rows, 105 Cols
VT100


mspdev35 - CRT
File Edit View Options Transfer Script Tools Window Help
RIB Diagnostic & Monitoring Tools
PUB/SUB Message Utilities Submenu
This Host: mspdev35
JMS Port : 36053
EHOME : /u01/rib11sys/egate
RDMTLOGS : /u01/rib11sys/egate/RDMTLOGS
RMS DB : suprtk11
SIM DB : suprtk11
RDM DB : rdm11
RPM DB : not used
RCOM DB : not used
AIP DB : not used
1 - Publish Msg Utility
2 - PUB Receipt test msg
3 -
4 -
5 - Subscribe Msg Utility
6 - SUB Receipt test msg
7 -
8 -
9 - JMS Topic Scan
10 -
11 -
12 - Hello World Test
99 - Main Menu
Selection:
Ready
Telnet
23, 12
32 Rows, 105 Cols
VT100


mspdev35 - CRT
File Edit View Options Transfer Script Tools Window Help
RIB Diagnostic & Monitoring Tools
e*ways Submenu
This Host: mspdev35
JMS Port : 36053
EHOME : /u01/rib11sys/egate
RDMTLOGS : /u01/rib11sys/egate/RDMTLOGS
RMS DB : suprtk11
SIM DB : suprtk11
RDM DB : rdm11
RPM DB : not used
RCOM DB : not used
AIP DB : not used
1 - Start an e*Ways
2 - Stop an e*Ways
3 - Verify e*way
4 - View e*way RIBLOG
5 - Set TRACE logging (SB)
6 - Set INFO logging (SB)
7 - Set NO logging (SB)
8 - View e*way SB Logfile
9 - View Egate.txt
10 - View e*way list (registry)
11 - create Egate.txt
12 - plist (eways)
99 - Main Menu
Selection:
Ready
Telnet
23, 12
36 Rows, 107 Cols
VT100


mspdev35 - CRT
File Edit View Options Transfer Script Tools Window Help
RIB Diagnostic & Monitoring Tools
Start/Stop Submenu
This Host: mspdev35
JMS Port : 36053
EHOME : /u01/rib11sys/egate
RDMTLOGS : /u01/rib11sys/egate/RDMTLOGS
RMS DB : suprtk11
SIM DB : suprtk11
RDM DB : rdm11
RPM DB : not used
RCOM DB : not used
AIP DB : not used
1 - Start ALL e*Ways
2 - Stop All e*Ways & JMS
3 -
4 - plist (eways)
5 - Stop JMS
6 - Start JMS
7 - stccmd
8 - plist (stcms)
9 - Stop Registry & CB
10 - Start Registry & CB
11 - Verify Registry & CB
12 - plist (all)
99 - Main Menu
Selection:
Ready
Telnet
23, 12
36 Rows, 107 Cols
VT100

Chapter 6 – Tools overview

RIB log scan

scan_logs.sh

scan_logs_delta.sh

loglookrib.sh

loglookrib_delta.sh

loglookrib_delta_mail.sh

loglookrib_mail.sh

loglooksb.sh

loglooksb_delta.sh

loglooksb_delta_mail.sh

loglooksb_mail.sh

These scripts perform a log scan to look for a /pattern/ (“Exception”) in all of the log files in a directory. It then writes the matches to a single log file. This becomes the base file. A second script (delta) looks for the same pattern, but compares the matches against the base file, and outputs only new ones. The primary scripts is the scan_logs.sh and the scan_logs_delta.sh. The others merely wrapper the calls based on log directory or if email notification in used.

The files created and used by these scripts are controlled by the rdmt.conf entries.

The location of these files should be sized to handle fairly large text files, since it is possible for there to be a lot of exceptions and these will contain the consolidated entries from potentially hundreds of logs.

The scan and the view have been separated. The scan produces the summary; the view will show the consolidated file. This was done to support rsh functionality.

Sample output:

```
Scanning Logs...04Sep27-1418
Test Pattern : Exception
logfiles path: /egate/RIBLOGS
Output file : /tmp/riberrors.txt
Number of Logs scanned: 82
Total number lines with pattern matches: 31990
Logs with error pattern Exception: 3
Number of matches per Log:
50:/egate/RIBLOGS/rib_ewASNInToRDMWH1.log
40:/egate/RIBLOGS/rib_ewOrderToRDMWH1.log
31900:/egate/RIBLOGS/rib_ewStockOrderToRDMWH1.log
```

Log file archive

logarch.sh

The RIB application and SeeBeyond generate log files on a per e*way basis. These need to be archived on a daily basis. This script is written so as to be executable both on-demand and via cron. There is a sample crontab script supplied in the rdmt directory that can be added to the egate user's crontab to run the script once per day.

The files created by these scripts are controlled by the rdmt.conf entries. This script can be run as often as desired. It uses the date timestamp to the minute to create the archive directory name.

Assumptions:

1 Run as owner of the files or has r/w privs. to create the directories and mv files

2 *In Releases RIB 10.3.3 and earlier -- The RIBLOGS logs are created at e*way start-up and are not currently re-created on the fly*, so this script copies them and then truncates the existing log by over-writing with /dev/null.

3 The SeeBeyond logs are located in the location created by the default install: $EHOME/client/logs.

RIB Hospital scans

htest.sh

htest_mail.sh

AccessHospital.class

These scripts use JDBC to access the database(s) containing the Hospital tables. They write the matches to a single log file. This becomes the base file. A second script (delta) looks for the same pattern, but compares the matches against the base file, and outputs only new ones.

They scan the Hospital and report thing such as

- how many messages (row count),
- how many have exceed the retry count,
- how many of a topic,
- and compares against a threshold value

Sample output:

```
Connect String: jdbc:oracle:thin:@hqibm43:1521:rdm2w
Database User : rdm10
RIB Hospital Access Test
Rib_message table row count : 5249
Messages at max_attempts    : 5247
JMS Topics and row count in Hospital:
9        etASNInFromRIBToWH1
3        etItemsTLFromRIB
15       etOrderFromRIBToWH1
5222     etStockOrdersFromRIBToWH1
End of Test...
Scan for Threshold...
WARNING -- Threshold Exceeded!  RDM Hospital Row Count:  5249
```

Hospital failure scan

htest_failures.sh

AccessHospitalFailures.class

These scripts scan the rib_messages hospital table and for each message it pulls the description from the rib_message_failure table.

Sample output:

```
Scanning Hospital Message Failures
Scanning Hospital...04Apr22-0913
Connect String: jdbc:oracle:thin:@salmerdbs1:1521:rtkm
Database User : rmsr
RIB Hospital Failures
Message #: 233422
Date     : 2004-04-19 06:13:33.0
Location : ewInvReqToRMS.colInvReqToRMS
java.sql.SQLException: OracleObjectSubscriberHelper.execute() threw
Exception: Error from {call RMSSUB_INVREQ.CONSUME(?,?,?,?)}: [E]
Error ORA-01407: cannot update ("RMS10"."ORDSKU"."LATEST_SHIP_DATE")
to NULL returned by program unit CREATE_ORDER_SQL.SET_ORDER_DATES.
Message #: 233423
Date     : 2004-04-19 06:18:14.0
Location : ewInvReqToRMS.colInvReqToRMS
```

```
java.sql.SQLException: OracleObjectSubscriberHelper.execute() threw
Exception: Error from {call RMSSUB_INVREQ.CONSUME(?,?,?,?)}: [E]
Error ORA-01407: cannot update ("RMS10"."ORDSKU"."LATEST_SHIP_DATE")
to NULL returned by program unit CREATE_ORDER_SQL.SET_ORDER_DATES.

Message #: 233424

Date     : 2004-04-19 06:21:19.0

Location : ewInvReqToRMS.colInvReqToRMS

java.sql.SQLException: OracleObjectSubscriberHelper.execute() threw
Exception: Error from {call RMSSUB_INVREQ.CONSUME(?,?,?,?)}: [E]
Country is not valid. Country Id: %s1

Message #: 233425

Date     : 2004-04-19 06:22:04.0

Location : ewInvReqToRMS.colInvReqToRMS

java.sql.SQLException: OracleObjectSubscriberHelper.execute() threw
Exception: Error from {call RMSSUB_INVREQ.CONSUME(?,?,?,?)}: [E]
Country is not valid. Country Id: %s1
```

RMS MFQ table scan

AccessTables.class

ttestrms.sh

This script uses JDBC to access the databases containing the RMS application MFQ tables.

It scans all of the RIB interface tables and reports how many messages (row count). The script reads a file called mfqtables.conf for the list of tables to scan.

Sample output:

```
Scanning MFQ Tables for All Non-zero Tables...


Connect String : jdbc:oracle:thin:@hqibm42:1521:rms2w

Database User  : rms

Table List File: /home/egate/rdmtXX/mfqtables.conf


Reading Conf File


**** Table & Row Count ***

Tablename ALLOC_MFQUEUE=8

Tablename ATP_MFQUEUE=1820791

Tablename ORDER_MFQUEUE=59


Scan for Threshold...

WARNING -- Threshold Exceeded! Tablename ATP_MFQUEUE=1820791

Enter Key to continue...
```

RDM upload table scan

AccessTables.class

ttestrms.sh

This script uses JDBC to access the databases containing the RDM application Upload tables.

It scans all of the RIB interface tables and reports how many messages (row count). The script reads a file called uploadtables.conf for the list of tables to scan.

Sample output:

```
Scanning Upload Tables for All Non-zero Tables...

Connect String : jdbc:oracle:thin:@hqibm43:1521:rdm2w
Database User  : rdm10
Table List File: /home/egate/rdmtXX/uploadtables.conf

Reading Conf File

**** Table & Row Count ***
Tablename INV_ADJUSTMENT_TO_UPLOAD=2

Scan for Threshold...

Enter Key to continue...
```

JMS topics scans

jmstopics.pl

jmstopics_scan.pl

These are Perl scripts that query the JMS Queue for all of the topics and the message count on each topic, or just the topics with message counts.

Sample output:

```
Scanning for Topics with message counts...05Feb22-1619
MSGCNT    TOPIC                SSEQ          ESEQ
============================================================
78        etItemLocFromRMS     283    360    THRESHOLD WARNING
65        etItemLocISO         65     129    THRESHOLD WARNING
3         etItemsISO           231    233    THRESHOLD WARNING
34        etMerchHierFromRMS   12     45     THRESHOLD WARNING
2         etStockOrdersISO     6      7      THRESHOLD WARNING
Enter Key to continue...
```

JMS tools

jms_status.sh

topicsubs.sh

deletemsg.sh

dmpmsg.sh

Dump JMS topic msg

This script will prompt for the JMS topic name and the message number, and will then dump a copy into an output file. The JMS message data is passed through a filter to convert the macros used to save space (&lt,&gt,&quot,&apros) to the more readable xml.

Sample output:

```
Dump JMS Message
JMS Port:  24053
JMS Topic Name : etVendorFromRMS
Message Number : 311
Output filename: /tmp/msg311.txt
Reading Message Number 311 from etVendorFromRMS into /tmp/msg311.txt
converting to msg to xml
```

```
<?xml version="1.0" encoding="UTF-
8"?><RibMessages><ribMessage><family>vendor</family><type>VendorHdrM
od</type><id>67429365</id><rib

messageID>10.3.1|ewVendorFromRMS|colVendorFromRMS|2004.03.16
07:54:28.228|1</ribmessageID><publishTime>2004-03-16 07:54:28.483
EST</

publishTime><hospitalID></hospitalID><messageData><!DOCTYPE
VendorHdrDesc SYSTEM
"http://www.retek.com/dtd/rib/VendorHdrDesc.dtd">

<VendorHdrDesc>

   <supplier>67429365</supplier>

   <sup_name>HANES HOSIERY INC.</sup_name>

   <contact_name>WENDY MOSELEY</contact_name>

   <contact_phone>8008761253 2080</contact_phone>

   <contact_fax>8043848616</contact_fax>

   <contact_pager/>

   <sup_status>A</sup_status>

   <qc_ind>N</qc_ind>

   <qc_pct/>

   <qc_freq/>

   <vc_ind>N</vc_ind>

   <vc_pct/>

   <vc_freq/>

   <currency_code>USD</currency_code>

   <lang>1</lang>

   <terms>A3</terms>

   <freight_terms>01</freight_terms>

   <ret_allow_ind>Y</ret_allow_ind>

   <ret_auth_req>N</ret_auth_req>

   <ret_min_dol_amt/>

   <ret_courier/>

   <handling_pct/>

   <edi_po_ind>N</edi_po_ind>

   <edi_po_chg>N</edi_po_chg>

   <edi_po_confirm>N</edi_po_confirm>

   <edi_asn>N</edi_asn>

   <edi_sales_rpt_freq>W</edi_sales_rpt_freq>

   <edi_supp_available_ind>N</edi_supp_available_ind>

   <edi_contract_ind>N</edi_contract_ind>
```

```
    <edi_invc_ind>Y</edi_invc_ind>
    <cost_chg_pct_var>0</cost_chg_pct_var>
    <cost_chg_amt_var>0</cost_chg_amt_var>
    <replen_approval_ind>N</replen_approval_ind>
    <ship_method>80</ship_method>
    <payment_method>OA</payment_method>
    <contact_telex/>
    <contact_email/>
    <settlement_code>N</settlement_code>
    <pre_mark_ind>N</pre_mark_ind>
    <auto_appr_invc_ind>N</auto_appr_invc_ind>
    <dbt_memo_code>Y</dbt_memo_code>
    <freight_charge_ind>N</freight_charge_ind>
    <auto_appr_dbt_memo_ind>N</auto_appr_dbt_memo_ind>
    <prepay_invc_ind>N</prepay_invc_ind>
    <backorder_ind>N</backorder_ind>
    <vat_region/>
    <inv_mgmt_lvl>S</inv_mgmt_lvl>
    <service_perf_req_ind>N</service_perf_req_ind>
    <invc_pay_loc/>
    <invc_receive_loc/>
    <addinvc_gross_net/>
    <delivery_policy>NEXT</delivery_policy>
    <comment_desc/>
    <default_item_lead_time>10</default_item_lead_time>
    <duns_number>067429365</duns_number>
    <duns_loc/>
    <bracket_costing_ind>N</bracket_costing_ind>
    <vmi_order_status/>
    <dsd_supplier_ind>N</dsd_supplier_ind>
</VendorHdrDesc>
</messageData><customFlag>F</customFlag></ribMessage></RibMessages>
Enter Key to continue...
```

JMS topics subscribers

This script will list all of the JMS Queue's subscribers and some basic stats about them.

Sample output:

```
Number Of Subscriber(s): 83
Subscriber name: NonDurable1
        Client ID:
        Topic name: STCMS.Control
        Message selector:
        Committed sequence: 859
        High sequence: 0
Subscriber name: etASNInFromRIBToISO:164-226-186-114:1077652163308
        Client ID: EWAY
        Topic name: etASNInFromRIBToISO
        Message selector: (retryLocation is null or retryLocation =
'Store1:RibMessagingComponent.ASNIn') and (groupKey is null or
groupKey in (''))
        Committed sequence: 0
        High sequence: 0
Subscriber name: etASNInFromRIBToISO:164-226-186-114:1078868399634
        Client ID: EWAY
        Topic name: etASNInFromRIBToISO
        Message selector: (retryLocation is null or retryLocation =
'Store1:RibMessagingComponent.ASNIn') and (groupKey is null or
groupKey in (''))
        Committed sequence: 1
        High sequence: 1
Subscriber name: etWHPhys:164-226-186-114:1077652163308
        Client ID: EWAY
        Topic name: etWHPhys
        Message selector: (retryLocation is null or retryLocation =
'Store1:RibMessagingComponent.WH') and (groupKey is null or groupKey
in (''))
        Committed sequence: 0
        High sequence: 0
```

JMS topic msg stats

This script will prompt for the JMS topic name and the number of messages to report on and will then dump a copy into an output file.

Sample output:

```
Dump JMS Message Stats
JMS Port:  25053
JMS Topic Name : etOrderISO
NUmber of Msg  : 10
Output filename: /tmp/msgstats.txt
Number Of Messages(s): 10
Message[1]:
Message.SeqNo=23
Message.Size=8037
Message.EnqueueTime=1078926077678
Message.JMSProperty.DM=1
Message.JMSProperty.DN=etOrderISO
Message.JMSProperty.EX=0
Message.JMSProperty.MI=ID:ad90d275:fb17790156:855e:a4e2ba65:fb35016e
e0
Message.JMSProperty.PR=4
Message.JMSProperty.RD=False
Message.JMSProperty.TS=1078926077664
Message.UserProperty.MESSAGE_LENGTH=7564
Message.UserProperty.PUBLISHER_NAME=colOrderToOrderISOFromRIB
Message.UserProperty.PUBLISHER_UUID={6799D93A-46E8-11D8-9093-
810453D9F5C5}
Message.UserProperty.marshalEncoding=UTF-8
Message.UserProperty.threadValue=1
```

```
Message[2]:
Message.SeqNo=24
Message.Size=6867
Message.EnqueueTime=1078926197769
Message.JMSProperty.DM=1
Message.JMSProperty.DN=etOrderISO
Message.JMSProperty.EX=0
Message.JMSProperty.MI=ID:ec1cf3f9:fb17790156:855e:a4e2ba65:fb350344
03
Message.JMSProperty.PR=4
Message.JMSProperty.RD=False
Message.JMSProperty.TS=1078926197762
Message.UserProperty.MESSAGE_LENGTH=6394
Message.UserProperty.PUBLISHER_NAME=colOrderToOrderISOFromRIB
Message.UserProperty.PUBLISHER_UUID={6799D93A-46E8-11D8-9093-
810453D9F5C5}
Message.UserProperty.marshalEncoding=UTF-8
Message.UserProperty.threadValue=1
```

JMS delete message(s)

This script will prompt for the JMS topic name and the beginning and ending sequence number of the messages to delete.

Sample output:

```
-------------------------------------------------
    Delete JMS Message Utility
-------------------------------------------------
 This utility was developed to delete message(s)
 on a JMS topic.

 WARNING!! message(s) are deleted!!  -- there is no
           recovery
```

```
 Usage:
       JMS Port
       topic name
       start sequence number
       ending sequence number

Enter to Continue? y/n/q [y]:
JMS Port:  36053
Topic Name: etReceiving
Start Message Seq Number: 5
End Message Seq Number  : 5
Enter to Continue? y/n/q [y]:
```

E*Way tools

ewayutil.sh

This utility tool was written to support the test and debug efforts and are focused on single e*way level of control. It wrapper the start and stop scripts as well as the stccmd utility to allow the setting of logging levels.

PUB/SUB tools

pubmsgutil.sh

pubmsg.sh

submsg.sh

submsgutil.sh

pubHelloWorld.sh

These tools are provided as examples to wrapper the java classes that are available to talk directly to the JMS. These simple wrappers provide basic test for the JMS level activity, including a simple round-trip test; HelloWorldTest that publishes and subscribes looking for the simple string “Hello World”.

Start and stop tools

startJMS.sh

stopJMS.sh

start_rib.sh

stop_rib.sh

These tools are provided as examples to wrapper to the start and stop scripts supplied by the RIB GA code, as well as the start_rib.sh script. Some also wrapper the SB stccmd utility, as well as a script that allows direct access to the commandline version; stccmd.

Export full schema

export_rib_schema.sh

This script will perform a full schema export on the unix system. It is similar to the GUI tool version, but places the files on the Unix system, and is much faster. The directory can then be tar'd. It is useful for version control as well as basic backups.

Assumptions:

The script is run as egate owner of the files or has r/w privs. to create the directories and mv files

This script expects a directory path. The path should already exist except for the last directory. The command will create that directory and then export to it. It must not already exist. Make sure the directory you specify has not been used for an export before!

Import full schema

import_rib_schema.sh

This is not provided as a Menu Item.

This script will perform a full schema import on the Unix system. It is similar to the GUI tool version, but pulls the files from a directory on the Unix system, and is much faster.

This will overwrite the RIBxxx Schema in the Registry. It does preserve the existing RIBxxx Schema by renaming it to -Prev.

Warning:

If there are messages on the JMS and in Hospital or XA logs, these could potentially become invalid, especially if there is a version difference in the e*Ways.

Assumptions:

1 Run as Administrator.

2 The Registry and Control Broker are down.

3 The user environment variables for EHOME, EGATE_SERVER_NAME, EGATE_SERVER_PORT are set appropriately.

Sample cron jobs

There are several cron job examples. These are written to call the tool scripts that have been modified to support cron. The only difference is the calling script has the PATH set, since cron jobs don't easily pickup environmental variables. All of these scripts have been tested using Appworx as well. Note that when using Appworx, sudo has to be setup so that the scripts execute as egate.

cron_scanlogs.sh

This cron jobs calls the scanlogs delta scripts. These are scripts that wrapper the basic scan logs for RIBLOGS and SB logs and test for exceptions, but only notify on deltas. An email is sent to the wmaillist.conf user addresses. See sample contab file

cron_scanhosp.sh

This cron jobs calls the scan hospital delta scripts. These are scripts that wrapper the basic scan hospital scripts for growth and thresholds, will only notify on deltas. An email is sent to the wmaillist.conf user addresses. See sample contab file

cron_logarch.sh

This cron jobs calls the logarch.sh script. It is useful in archiving the logs every night. See sample contab file

cron_ribhealth.sh

Verifies the SB process and executes many of the tools to provide a snapshot of the state of health. The logs are kept in ~/RDMTLOGS.

cron_procmem.sh

Verifies the SB processes memory utilization against the threshold values. Will email to wmaillist user addresses.

Sample crontab:

45 23 * * 0-6 /egate/453/rdmtXX/cron_logarch.sh 1>/dev/null 2>&1

0,15,30,45 * * * 0-6 /egate/453/rdmtXX/cron_scanlogs.sh 1>/dev/null 2>&1

0,15,30,45 * * * 0-6 /egate/453/rdmtXX/cron_scanhosp.sh 1>/dev/null 2>&1

0,3045 * * * 0-6 /egate/453/rdmtXX/cron_ribhealth.sh 1>/dev/null 2>&1

This has the archiver run every night at 23:45, and the log and hospital scans run every 15 minutes 24x7. The cron_ribhealth scripts run every 3 minutes.

The scripts called use the rdmtmailer to send output to the email list (maillist.conf). They are set to send "No Change" messages currently.

start_rib.sh

The start_rib.sh and stop_rib.sh scripts are re-writes of the scripts shipped with the GA version of the RIB. These scripts use the Egate.txt file format to handle the starting and stopping of the RIB in the sequence guaranteed to prevent selector issues.

The reason for the re-write was to speed up the execution, as well as separate the functionality into switch controllable tasks, and to add some additional capabilities such as verification only process, as well as understanding of the Alert Agent. In addition, the scripts contain several environment variables to allow execution by cron or appworx.

Start_rib.sh was enhanced to be smarter about the Registry, Control Broker and JMS processes, and now has a verify switch that actively interrogate all components for a status response.

cron_ribhealth.sh uses the enhanced version of start_rib.sh and the verify switch (-v), to actively interrogate all of the rib components.

USAGE start_rib.sh: [-v][-r][-s schema_name][-f eway_file][-u user_name][-p user_password][-e eway_name][ALL] [SUB] [TAFR] [PUB] [HOSP]

Tests: Tests for running Registry, Control Broker, and JMS

Where:

-v	verify only -- creates a "failed_eways.txt" file with the names not booted
-r	specifies to verify -- creates a "failed_eways.txt" file with the names not booted
-s schema_name	specifies the name of the schema to start -- default is RIB103
-f eway_file	specifies the file containing e*way description, default is /egate/Egate.txt
-u user_name	specifies the user name to use -- default is Administrator
-p user_password	specifies the password to use -- default is STC
-e eway_name	specifies only a single e*way to start
ALL	specifies bringing up all elements listed in the e*way file. Equivalent to SUB TAFR PUB HOSP
SUB	specifies bringing up all SUB (subscriber) elements listed in the e*way file
TAFR	specifies bringing up all TAFR elements listed in the e*way file
PUB	specifies bringing up all PUB (publisher) elements listed in the e*way file
HOSP	specifies bringing up all HOSP (hospital) elements listed in the e*way file

BRDG	specifies bringing up all BRDG (bridge) elements listed in the e*way file
ONE	specifies bringing up one element

The script environment variables are set initially by the installation setup.sh scripts. But may need to be edited for each system, such as when a password change happens, unless the registry and control broker have been configured to use the –acl and you have create the .egate.stcpass file. They should always be owned by egate and permissions set to 700.

They use a file Egate.txt that is expected in $EHOME. The –f switch allows alternate files and locations.

Typical usage:

- start_rib.sh ALL

This will start ALL components in the Egate.txt located in the $EHOME directory. If the ALERT_AGENT=true -- then it will also try and start the agent.

The script issues the start cmd for each e*way. It first tests for a running Registry, Control Broker and JMS, if not found it will attempt to start them. Failure will force an exit and the script will not continue.

- start_rib.sh [SUB] [PUB] [TAFR] [HOSP] [BRDG]

Will start just that category of e*way in the Egate.txt file

- start_rib.sh –f /egate/Egate_batch.txt ALL [SUB] [PUB] [TAFR] [HOSP] [BRDG]

Will start using an alternate file.

- start_rib.sh –r ALL
- start_rib.sh –r -f /egate/Egate_batch.txt ALL [SUB] [PUB] [TAFR] [HOSP] [BRDG]

Will issue start commands on the first pass, then perform a second validation pass. Any e*ways not responding as “up” will be added to file: failed_eways.txt in the $EHOME directory. This file is in the format of an Egate.txt and can be re-submitted to the start_rib.sh using the –f switch.

- start_rib.sh –v ALL [SUB] [PUB] [TAFR] [HOSP] [BRDG]
- start_rib.sh –v -f /egate/Egate_batch.txt ALL [SUB] [PUB] [TAFR] [HOSP] [BRDG]

Will perform a validation pass ONLY. Any e*ways not responding as “up” will be added to file: failed_eways.txt in the $EHOME directory. This file is in the format of an Egate.txt and can be re-submitted to the start_rib.sh using the –f switch.

-]start_rib.sh –e ewHospitalRetryRMS

Will start just that e*way.

stop_rib.sh

USAGE stop_rib.sh: [-r][-s schema_name][-f eway_file][-u user_name][-p user_password][-e eway_name][ALL] [SUB] [TAFR] [PUB] [HOSP]

Tests: Tests for running Registry and Control Broker

Where:

-r	specifies to verify -- creates a "failed_eways.txt" file with the names not stopped
-s schema_name	specifies the name of the schema to stop -- default is RIB103
-f eway_file	specifies the file containing eway description, default is $EHOME/Egate.txt
-u user_name	specifies the user name to use -- default is Administrator
-p user_password	specifies the password to use -- default is STC
-e eway_name	specifies only a single eway to stop
ALL	specifies stopping all elements listed in the eway file. Equivalent to SUB TAFR PUB HOSP
SUB	specifies stopping all SUB (subscriber) elements listed in the eway file
TAFR	specifies stopping all TAFR elements listed in the eway file
PUB	specifies stopping all PUB (publisher) elements listed in the eway file
HOSP	specifies stopping all HOSP (hospital) elements listed in the eway file
BRDG	specifies stopping all BRDG (bridge) elements listed in the e*way file
ONE	specifies stopping one element

Typical usage:

- stop_rib.sh ALL

This will stop ALL components via a stopall command to the Control Broker.. If the ALERT_AGENT=true -- then it will also try and stop the agent.

It first tests for a running Registry and Control Broker -- failure will force an exit and the script will not continue.

- stop_rib.sh [SUB] [PUB] [TAFR] [HOSP]] [BRDG]

Will stop just that category of e*way in the Egate.txt file

- stop_rib.sh –f /egate/Egate_batch.txt ALL [SUB] [PUB] [TAFR] [HOSP] [BRDG]

Will stop using an alternate file.

cron_ribhealth.sh

This cron script is written to take a snapshot of the RIB status as well as the system status at a point in time by calling several of the tool kit utilities as well as Unix level utilities. It generates a log file that is stored in the RDMTLOGS directory. This log file is time stamped.

rh-05Feb03-0907 rh-05Feb13-1432 rh-05Feb13-1454 rh-05Feb13-1506

rh-05Feb13-1516 rh-05Feb14-1148 lastrun-mspdev35

The script also continually overwrites the lastrun file, to make it easy to look at the last data without scanning thru logs.

There is a script and a conf file include in the tools:

ftp_rhlog.sh loghost.conf

That will do log forwarding of this file to a central host.

The cron_ribhealth.sh script produces output to the log file, but then parses the log looking for the keywords that the tools have been written to output: DOWN, FAIL and WARNING. The script will then call the mailer the appropriate email list.

The script was written to run as an Appworx job as well.

RIBConfigReport.sh

This script was written to take a snapshot of the RIB environment and test for some basic configuration issue; such as mismatched jar files between client and server directories. It will also attempt to compare between the INSTALL directories as well.

The script is stand-a-lone, similar to a cron job, but expects nothing from the tool kit configuration files or other tools in the tool kit directory.

The report output is two-fold; a summary to the user of PASS/FAIL, as well as details in the RIBConfigReport.out file.

USAGE

RIBConfigReport.sh -s schema -i retek_install_dir -o outfile

Where:

Default is RIB_SCHEMA and RETEK_INSTALL_DIR values set in egate_profile

-s rib_schema	is the SeeBeyond Schema name
-i retek_install_dir	is the directory where the RIB tar files were expanded to.
-o outfile	is the full path to output file (/tmp/RIBConfigReport.out is default

Manditory:

EHOME must be set to SeeBeyond egate root

RIB_SCHEMA must be set to desired schema to report against.

Optional:

RETEK_INSTALL_DIR set to location RIB was installed from.

Chapter 7 –Tool usage examples

How do I

“The RIB is down! – How do I know if it is up?”

1 Using the menu system, select the RIB Health submenu

2 Select - Verify SB Only (REG, CB, JMS)

3 This will verify that the basic SeeBeyond process are there and responding

4 If not then proceed to Start/Stop Submenu and select - Start Registry & CB

   a Proceed to Start/Stop Submenu and select - Start Registry & CB

   b Select - Start JMS

5 If there Select - Verify e*Ways. This will verify that the e*ways listed in the Egate.txt are up and responding.

“How do I know if the RIB install is correct?”

1 Install RIB Diagnostic and Monitoring toolkit. Use the same configuration parameters that are used in the configuration of the RIB (See RIB Installation Guide). This will insure that the tool kit is configured to test and diagnose appropriately, as well as help determine if the RIB parameters are accurate.

2 Using the menu system, select the RIB Health submenu

3 Select - Execute RIB Config Report. This will produce a basic report on the installation.

4 Select - View Config Report output to review installation.

5 Return to main menu and select e*way Submenu

6 Select - create Egate.txt. This will create the correct Egate.txt file from the installed e*ways in the registry.

7 Move new Egate.txt to $EHOME. Edit as desired for test and production.

8 Start the e*way, either using the GA start_rib script in $EHOME or

   a Using the menu system, select the Start/Stop Submenu.

   b Select - Start ALL e*Ways

9 Select - Verify e*Ways. This will verify that the e*ways listed in the Egate.txt are up and responding.

"I installed the RIB – How do I know it works?"

1 Follow the RIB Installation Guide.

2 Install the RIB RIB Diagnostic and Monitoring toolkit.

3 Create the correct Egate.txt file.

4 Run the steps in "How do I know if the RIB install is correct?"

5 Select main menu

6 Select RIBLOGS Logs Scan

7 Select SB Logs Scan

"How do I know where my issue is occurring?"

1 Select main menu

2 Select RIBLOGS Logs Scan. Look for error

3 Select SB Logs Scan Look for errors

4 Select JMS Topic Scan. Look for topics with messages stuck

5 Using the menu system, select the RIB Health submenu. Select Verify e*Ways (Egate.txt). Look for DOWN or FAIL

6 Select - Execute rib_health script. Then Select View lastrun cron_ribhealth and look for DOWN, FAIL or WARNING.

"How do I test an Interface (e*way)?"

Using the menu system, select the e*way submenu. Use the individual control selections to test the interface.

"How do I monitor the RIB in production?"

There are several example cron jobs that are developed and supplied to run and provide basic monitoring and email notification. See the section on cron jobs for details.

"I want to know how to use SeeBeyond commands (stcxxx)"

Many of the scripts in the tool kit merely wrapper the SeeBeyond commandline utilities. They can be looked at for examples. Almost all of the JMS tools use the SB utilities.