Nortel CDMA Gateway User Guide

Gateway Release: 3.4.0

Date: 23 November 2007

Contents

References

Name	Description
Gateway Framework User Guide	This user guide describes in detail the functionality of the Gateway Framework, and the standard suite of tools available.
CDMA System Performance Guide	CDMA System Performance – Operational Measurements Guide 411-2133-525
DMS-MTX Operational Measurements Reference Manual	This user guide defines the operational measurement (OM) groups and registers applicable only to a DMS-MTX cellular telephone system. – 411-2131-814.15.17 - September 2005
Nortel DO-EMS User Guide	Nortel CDMA2000 1xEV-DO Element Management Subsystem (DO-EMS) User Guide – 411-2133-927.01.18 - Standard April 2004

Glossary

PIF	Parser Intermediate Format
LIF	Loader Input Format
1xEVDO	1X Evolution Data Only.
BSS	Base Station Subsystem
BSC	Base Station Controller
RNC	Radio Network Controller
DOM	Data Only Module

Preface

About this Guide

This guide details the Vendor specific information on the Gateway release for Nortel CDMA. It contains the following information:

- *Chapter 1. Overview.* This chapter gives a brief description of the Vendor Gateway and the raw data format it parses.
- *Chapter 2: Engine Rules and Configuration.* This chapter details the vendor specific rules for parsing the raw data and their configuration.
- *Chapter 3: Post Parser Rules and Configuration.* This chapter describes any vendor specific Post Parser rules and their configuration.
- *Chapter 4: Tech Pack Support.* This chapter describes any standard support for Tech Packs included with the Gateway.
- *Chapter 4: Installation specific information.* This chapter contains the customer installation specific information.

Conventions

The following conventions are used in this guide:

`fixed width`	Highlights a block of example code, a configuration entry, or a command line instruction

1. Overview

1.1 The Gateway Framework

The Nortel CDMA Gateway, referred to as the Vendor Gateway, uses the Gateway Framework as a container for the execution of its engine and post parser stages. The Gateway Framework and Vendor Gateway are decoupled into two separate installations. The Gateway Framework consists of a library of Perl modules that provide functionality such as:

- a container for the execution of the Vendor Engine and Post Parser rules for data transformation
- Intermediate (PIF) and output data (LIF) storage and management
- logging utilities
- cleanup and crash recovery
- statistics gathering

The Vendor Gateway plugs into the Gateway Framework and extends this functionality to provide the final Gateway that parses the vendor data.

More information on the standard Gateway configuration is contained in the Gateway Framework User Guide.

Only vendor specific configuration details will be described in this document.

1.2 Nortel CDMA Gateway Overview

1.2.1 Network Details

The Nortel CDMA performance and configurations are obtained for the following network elements:

- Nortel MTX
- Nortel EVDO
- Nortel RAN
- Nortel PDSN

1.2.2 DMS Operational Measurements

Operational Measurements (OMs) are counts of call processing events that provide the operating company with performance data for the DMS-MTX system. System engineers use OM records to ensure that the DMS-MTX switch operates efficiently. OMs provide information regarding system performance, grade of service being offered, connecting facilities, performance, and traffic levels of various elements internal to and connected to the system.

Each time a call is initiated, handed off, or terminated, different call processing events take place within the DMS-MTX system. The basic operation of the Operational Measurement (OM) subsystem is to monitor and count these events during a given transfer period. The results, known as operational measurements (OMs), are then recorded into a set of active registers in the data store.

1.2.3 SDM

The SuperNode Data Manager (SDM) is a fault-tolerant UNIX-based processing platform that uses Motorola technology, and runs operations, administration, maintenance, and provisioning (OAM&P) software applications. It is a high-performance computing platform connected to the operating company's DMS switch.

The SDM and its applications allow the operating company to off-load its OAM&P processes from the switch. The SDM is connected to the operating company's network through an Ethernet connection to its operations intranet.

1.2.4 SDM File Transfer

File transfer can be accomplished with FTP, Secure File Transfer (SFT), or File Transfer Protocol Proxy (FTPP). In an SFT session, the SFT client can access either the SDM, or the computing module (CM) for the purpose of doing file transfers. A DCE account and password is required to use SFT.

The non-DCE SFT configuration uses standard file transfer protocol (FTP) to send un-encrypted (ASCII text) login UserIDs and passwords across the network from the FTP client to the SFT server.

1.2.5 Nortel MTX

The Nortel MTX performance data are obtained from the SDM, while the configuration data are obtained using the Virtuo DA (Data Acquisition) toolkit from the MTX.

1.2.5.1 Performance Data Layout

The performance data is known as data collection files. Currently the vendor gateway only supports collection of one file per MSC per period from the SDM.

Below is a sample of performance data:

```
Date, Time, Switch Name, Group Name, Key/Info Field, Reg1 Name, Reg1 Value, Reg2
Name, Reg2 Value, Reg3 Name, Reg3 Value, Reg4 Name, Reg4 Value, Reg5 Name, Reg5
Value, Reg6 Name, Reg6 Value, Reg7 Name, Reg7 Value, Reg8 Name, Reg8 Value, Reg9
Name, Reg9 Value, Reg10 Name, Reg10 Value, Reg11 Name, Reg11 Value, Reg12 Name,
Reg12 Value, Reg13 Name, Reg13 Value, Reg14 Name, Reg14 Value, Reg15 Name, Reg15
Value, Reg16 Name, Reg16 Value, Reg17 Name, Reg17 Value, Reg18 Name, Reg18
Value, Reg19 Name, Reg19 Value, Reg20 Name, Reg20 Value, Reg21 Name, Reg21
Value, Reg22 Name, Reg22 Value, Reg23 Name, Reg23 Value, Reg24 Name, Reg24
Value, Reg25 Name, Reg25 Value, Reg26 Name, Reg26 Value, Reg27 Name, Reg27
Value, Reg28 Name, Reg28 Value, Reg29 Name, Reg29 Value, Reg30 Name, Reg30
Value, Reg31 Name, Reg31 Value, Reg32 Name, Reg32 Value

08-10-
2006,04:00:02,SNJNPRLGCM1,TRK,BSC01_SBS085.2W.192.192,INCATOT,42,PRERTEAB,0,INFA
IL,1,NATTMPT,26,NOVFLATB,0,GLARE,0,OUTFAIL,1,DEFLDCA,0,DREU,0,PREU,0,TRU,271,SBU
,0,MBU,0,OUTMTCHF,0,CONNECT,26,TANDEM,45,AOF,0,ANF,0,TOTU,271,ANSWER,16,INVAUTH,
0,BLKCTRK,0,MAXBU,0,TRU2WIN,0,NCTPASS,0,NCTFAIL,0,ACCCONG,0,NOANSWER,0,NPQUERY,4
3,NPRESP,43,NPBDRTF,0

08-10-
2006,04:00:02,SNJNPRLGCM1,TRK,BSC01_SBS086.TM2.192,INCATOT,30,PRERTEAB,0,INFAIL,
0,NATTMPT,16,NOVFLATB,0,GLARE,0,OUTFAIL,0,DEFLDCA,0,DREU,0,PREU,0,TRU,163,SBU,0,
MBU,0,OUTMTCHF,0,CONNECT,16,TANDEM,31,AOF,0,ANF,0,TOTU,163,ANSWER,13,INVAUTH,0,B
LKCTRK,0,MAXBU,0,TRU2WIN,0,NCTPASS,0,NCTFAIL,0,ACCCONG,0,NOANSWER,0,NPQUERY,29,N
PRESP,29,NPBDRTF,0

08-10-
2006,04:00:02,SNJNPRLGCM1,TRK,JACK.TM2.2,INCATOT,0,PRERTEAB,0,INFAIL,0,NATTMPT,0
,NOVFLATB,0,GLARE,0,OUTFAIL,0,DEFLDCA,0,DREU,0,PREU,0,TRU,0,SBU,0,MBU,0,OUTMTCHF
,0,CONNECT,0,TANDEM,0,AOF,0,ANF,0,TOTU,0,ANSWER,0,INVAUTH,0,BLKCTRK,0,MAXBU,0,TR
U2WIN,0,NCTPASS,0,NCTFAIL,0,ACCCONG,0,NOAN
```

Example 1:

In Example 1, the values exist for the file header, column-head and data fields. The values for the header fields are described in Table 1.

Header Field	***Value***	***Meaning***
Date	08-10-2006	Date of the data
Time	04:00:02	Time of the data
Switch Name	SNJNPRLGCM1	MSC/MTX's name
Group Name	TRK	OM group name
Key/Info Field	BSC01_SBS085.2W.192.192	Key or information field

Table 1 Header Fields

Column	***Description***	***Value in the example***
Reg1 Name	Measurement name	INCATOT
Reg1 Value	Measurement value	42

Table 2 Record Header Fields

1.2.5.2 Configuration Table Data Layout

The configuration data is used to do data lookup for the FDN construction. It consists of several configuration tables which are CLLI, CDMACELL, NOISLPP, SBSINV, C7LKSET, NOISBSC. A separate expect script module is used to gather this data from the SDM.

Below is a sample of MTX table data:

```
TABLE: C7RTESET

TOP

        ROUTESET   NETNAME            TFPBCAST                           DPC

                                                                      ROUTES

----------------------------------------------------------------------------
```

```
          CAGSTP   PR_MSC3                    N      ANSI7 (  2) ( 46) (  2)$
                                                     (CAGSTP  0) (SJ_STP  1)$
          SJ_STP   PR_MSC3                    N      ANSI7 (  2) ( 46) (  1)$
BOTTOM


TABLE: CLLI
TOP
            CLLI ADNUM TRKGRSIZ                         ADMININF
----------------------------------------------------------------
         DMODEMC      2        4             NEW_MODEM_3X02CA_CLLI
  DUMPANDRESTORE     28        0                   DUMP_AND_RESTORE
        OFFHKSUP     10       10       SUPERVISION_SIGNAL_OFFHOOK
         WINKSUP     20       10          SUPERVISION_SIGNAL_WINK
         ONHKSUP     11       10        SUPERVISION_SIGNAL_ONHOOK
         TRKLPBK     24        0                  TRUNK_LOOP_BACK
BOTTOM
```

1.2.5.3 File naming specification

The file naming convention for performance and the configuration data is as follows:

- Performance data:

Source file Naming Convention
<MSCId>.<FileCreationDate>.<FileCreationTime>.<SwitchName>. <ReportingInterval>.csv

<MSCId> is the MSC ID.

<FileCreationDate>.<FileCreationTime> is shortly after the data collection end-time.

<SwitchName> indicates the source MSC

<ReportingInterval> is usually "OT" indicating Office Transfer period.

- Configuration data:

The file naming convention for the configuration file is:

`mtxid.Tables.YYYYMMDDHHMM.cfg`

1.2.6 Nortel EVDO

The DO-EMS provides 1xEVDO data records and radio network configuration parameters reports in an ASCII readable format. The DO-EMS produces data using a defined directory structure and file naming convention.

Below provides examples of the data files. For a more detailed discussion the reader should refer to the Nortel documentation.

1.2.6.1 Performance Data Layout

The performance data is known as data collection files, dcFiles, which are located in the WebNMS home directory of the DO-EMS server.

Below is a sample dcFiles:

```
NodeName=RN_10.10.50.59

Category=Abis_InterfaceTraffic

ConfName=AbisTraffic

OIDs(Variables):

Var-1 = 1.3.6.1.4.1.6587.4.30.1.4.1.6(airAbisPeerMsgsSent)

Var-2 = 1.3.6.1.4.1.6587.4.30.1.4.1.5(airAbisPeerHellosReceived)

Var-3 = 1.3.6.1.4.1.6587.4.30.1.4.1.4(airAbisPeerHellosSent)

Var-4 = 1.3.6.1.4.1.6587.4.30.1.4.1.7(airAbisPeerMsgsReceived)


IndexOids:

1.3.6.1.4.1.6587.4.30.1.4.1.1(airAbisPeerIPAddr)

airAbisPeerIPAddr   Time                      Var-1  Var-2  Var-3  Var-4

--------------------------------------------------------------------------

10.10.50.1          02/05/2003 23:57:44       *N/A   N/A    N/A    N/A

10.10.50.1          02/05/2003 23:58:45       N/A    N/A    N/A    N/A

10.10.50.1          02/05/2003 23:59:44       N/A    N/A    N/A    N/A

10.10.50.1          02/06/2003 00:00:46       N/A    N/A    N/A    N/A

*N/A: “not available”
```

Example 2:

In Example 2, the values exist for the dcFile header, column-head and data fields. The values for the header fields are described in Table 1.

Header Field	***Value***	***Meaning***
NodeName	`RN_10.10.50.59`	Node on which the data is collected
Category	`Abis_InterfaceTraffic`	OM template name
ConfName	`AbisTraffic`	Configuration name (User-specified)
OIDs (Variables): Var-1	`1.3.6.1.4.1.6587.4.30.1.4.1.6(airAbnisPeerMsgsSent)`	Variable alias 1
OIDs (Variables):	`1.3.6.1.4.1.6587.4.30.1.4.1.5(airAbnisPeerHellosReceived)`	Variable alias 2

Var-2		
OIDs (Variables): Var-3	`1.3.6.1.4.1.6587.4.30.1.4.1.4(airAbnisPeerHellosSent)`	Variable alias 3
OIDs (Variables): Var-4	`1.3.6.1.4.1.6587.4.30.1.4.1.7(airAbnisPeerMsgsReceived)`	Variable alias 4
IndexOIDs	`1.3.6.1.4.1.6587.4.30.1.4.1.1(airAbnisPeerIPAddr)`	Instance variable

Table 3 Header Fields

Column	***Description***	***Value in the example***
Instance	The index OID variable	1.3.6.1.4.1.6587.4.30.1.4.1.1(airAbisPeerIPAddr)
Time	The timestamp of each sampling	The date and time of the timestamp - for example 02/10/2003 15:10:00 in the first row of the data values
Var-1	Variable alias 1	1.3.6.1.4.1.6587.4.30.1.4.1.5(airAbnisPeerHellosReceived) - for example, N/A in the first row of the data values
Var-2	Variable alias 2	1.3.6.1.4.1.6587.4.30.1.4.1.4(airAbnisPeerHellosSent) - for example, N/A in the first row of the data values
Var-3	Variable alias 3	1.3.6.1.4.1.6587.4.30.1.4.1.7(airAbnisPeerMsgsReceived) - for example, N/A in the first row of the data values
Var-4	Variable alias 4	1.3.6.1.4.1.6587.4.30.1.4.1.1(airAbnisPeerIPAddr) - for example, N/A in the first row of the data values

Table 4 Record Header Fields

1.2.6.2 Configuration Data Layout

There are 4 configuration data files.

- DOM
- INTERFACE
- IS856CHANNELELEMENT
- PDSN

1.2.6.3 File naming specification

The file naming convention for performance and the configuration data is as follows:

- Performance data:

The directory structure for the dcFiles is as follows:

`“<EMSdcDirectory>/<Nodename>/<date>/<DC_Templ_ConfNam>.dat”`

where **`<date>`** format is **`mmddyyy`**.

In Example 1 above, the directory structure would be:

`dcFiles/RN_10.10.50.59/02112003/Abis_InterfaceTraffic_AbisTraffic.dat`

- Configuration data:

The file naming convention for the 4 configuration files are:

domQueryMMDDYYYY_HHMM.txt

ifQueryMMDDYYYY_HHMM.txt

pnQueryMMDDYYYY_HHMM.txt

pdsnQueryMMDDYYYY_HHMM.txt

Every RNC consists of a set of configuration files. Every set of configuration files from the RNC will be transferred into its respective directory using the RNC ID:

dcFiles /<configuration_file>

1.2.7 Nortel RAN

In the Nortel NBSS PM system, counters are grouped by Operational Management groups (OM groups). The counter performance statistics from each OM is collected from the network subsystem (e.g. BTS) and sent to BSSM Manager using an event trigger.

These binary files will be processed by an Nortel internal OM parser to produce raw ASCII text files and available for upload and further processing by the vendor gateway through a standard output directory.

1.2.7.1 Performance Data Layout

The raw ASCII text files available for processing as described in 1.1.1 above are available in 2 formats:

- 'raw text' format
- 'logfiltered text' format

The vendor gateway was designed to process the 'raw text' format. The OMs or performance counter groups are identified by the FDN (Fully Distinguished Name) in the header information of the file followed by the OM Sequence Number (s) within the data section in the file.

The Sequence Numbers are located in the second column of the data row which itself consists of 3 columns of values.

- 1st column – Data type of the value
- 2nd column – Sequence Number
- 3rd column – Counter value which may consists of a single value or multiple number array containing values.

It is the sequence number that represents the particular PM counters as described in the document in section 1.1.4 above (under BSC/BSC OM descriptions).

To illustrate the various types of data which need to be processed, in the OM example below, the performance data log is identified as belonging to the **ADVANCEDSECTOR** MO by its FDN of *"O%:CCLN-1-CBS1:CELLS-1-CELLS1:MCBTS-1-MC1900BTS1341:MCBTSSUBSYSTEM-1-MCBTSSUBSYSTEM1:ROOT-1-ROOT1:BTSCALLPROCESSING-1-BTSCALLPROCESSING1:ADVANCEDFA-1-ADVANCEDFA1:ADVANCEDSECTOR-1-ADVANCEDSECTOR2"*.

```
|O%:CCLN-1-CBS1:CELLS-1-CELLS1:MCBTS-1-MC1900BTS1341:MCBTSSUBSYSTEM-1-
MCBTSSUBSYSTEM1:ROOT-1-ROOT1:BTSCALLPROCESSING-1-
BTSCALLPROCESSING1:ADVANCEDFA-1-ADVANCEDFA1:ADVANCEDSECTOR-1-
ADVANCEDSECTOR2
|7|48|97877
|129|97|5 0 0 2287 3447 4154
|1025|70|4 |0|0|0 |0|0|0 |0|0|0 |0|0|0
```

Table 5 NBSS PM records

The first row of data is found to be "**|7|48|97877**". Therefore, the Sequence Number is "48", and the data value (i.e., the number of pegs) attributable to that OM is "97877". (In this first example, the Sequence Number has been underlined for clarity.)

The 2nd row marked in ***italics*** (for clarity) refers to the OM sequence number 97 followed by an array of data containing 6 values (marked by an initial value of 6) followed by 6 peg values.

Another example of the OM data that can be found is shown on the 3rd row which consist of a complex array type. Here the OM sequence number is 70. The following digit '4' signifies 4 data values each data point consists of an array of 3 values.

As explained above, the sequence number represents the measured counter described in the vendor documentation. Here:

Sequence No. 48 – OverheadForwardLinkUtilUWAvg

Sequence No. 97 – VoiceFchForwardLinkUtilAverage

Sequence No. 70 – PerRingAccessCounts (Attempts, Successes, Failures)

1.2.7.2 Configuration Data Layout

The BSM MAP configuration files are network topology files that will be extracted from the EMS using the Virtuo DA. These files contain the topology data for the corresponding Nortel RAN performance data.

Below is a sample BSM MAP records:

```
#
"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA3"#    CDMA_FREQ = 475
"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA1"#    CDMA_FREQ = 425

"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA2"#    CDMA_FREQ = 450



"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA3:AdvancedSector1"#    SectorId = Alpha, #    SectorCellId
= 193

"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA3:AdvancedSector3"#    SectorId = Gamma, #    SectorCellId
= 195
"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA3:AdvancedSector2"#    SectorId = Beta, #    SectorCellId
= 194
```

```
"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA1:AdvancedSector2"#    SectorId = Beta, #    SectorCellId
= 194

"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA1:AdvancedSector1"#    SectorId = Alpha, #    SectorCellId
= 193
"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA1:AdvancedSector3"#    SectorId = Gamma, #    SectorCellId
= 195
"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA2:AdvancedSector2"#    SectorId = Beta, #    SectorCellId
= 194

"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA2:AdvancedSector1"#    SectorId = Alpha, #    SectorCellId
= 193
"O%:CBS1:Cells1:Compact1900BTS1012:MCBTSSubsystem1:Root1:BTSCallProcessi
ng1:AdvancedFA2:AdvancedSector3"#    SectorId = Gamma, #    SectorCellId
= 195
```

Table 6 BSM MAP records

1.2.7.3 File naming specification

The file naming for the configuration data is as follows:

These files only define the BTS hierarchy. The association with the MTX is obtained from the filename:

MTX.**3.0**.200606260230.BSMmap

1.2.8 Nortel PDSN

The Nortel PDSN system produces ACII Schema files and ACII Performance data files. Counter names are defined and mapped within the Schema files while the corresponding counter performance statistics are produced in the Performance data files.

The parser maps the counter names to the raw counter values using unique mapping keys present in both the Schema and Performance files.

1.2.8.1 Schema layout

The Schema should be configured to have the following format:-

<schema_type> *schema_name* <schema_format>

Below are sample Schema records:

```
system       systemsch                EMS,PDSNSystem,%date%,%time%,%sess-
ttlarrived%,%sess-ttlrejected%,%sess-ttlconnected%,%sess-ttlauthsucc%,%sess-
ttlauthfail%,%sess-ttllcpup%,%sess-ttlipcpup%,%sess-ttlsrcviol%,%sess-ttlkeepfail%,%sess-
curttlcalls%,%sess-cursipconn%,%sess-curmipconn%,%sess-curactcall%,%sess-
curdormcall%,%sess-curarrived%,%sess-curlcpnegot%,%sess-curlcpup%,%sess-curauth%,%sess-
curauthed%,%sess-curipcpup%,%sess-curdisc%,%a11-ttlarrived%,%a11-ttlrejected%,%a11-
ttldemult%,%a11-ttldereg%,%a11-curactive%,%fa-ttlarrived%,%fa-ttlrejected%,%fa-
ttldemult%,%fa-ttldereg%,%fa-curactive%,%ha-ttlarrived%,%ha-ttlrejected%,%ha-
ttldemult%,%ha-ttldereg%,%ha-curactive%,%sess-calldur-1min%,%sess-calldur-2min%,%sess-
calldur-5min%,%sess-calldur-15min%,%sess-calldur-1hour%,%sess-calldur-4hour%,%sess-
calldur-12hour%,%sess-calldur-24hour%,%sess-calldur-over24hour%,%sess-setuptime-
```

```
100ms%,%sess-setuptime-200ms%,%sess-setuptime-300ms%,%sess-setuptime-400ms%,%sess-
setuptime-500ms%,%sess-setuptime-600ms%,%sess-setuptime-700ms%,%sess-setuptime-
800ms%,%sess-setuptime-900ms%,%sess-setuptime-1sec%,%sess-setuptime-2sec%,%sess-setuptime-
3sec%,%sess-setuptime-4sec%,%sess-setuptime-6sec%,%sess-setuptime-8sec%,%sess-setuptime-
10sec%,%sess-setuptime-12sec%,%sess-setuptime-14sec%,%sess-setuptime-16sec%,%sess-
setuptime-
over16sec%,%incremental%,%enddate%,%endtime%,%localenddate%,%localendtime%,%sess-
ttlfailed%,%uptime%,%uptimestr%,%lic-pdsn%,%lic-ha%,%lic-ggsn%,%lic-l2tplns%,%sess-
txbytes%,%sess-rxbytes%,%sess-txpackets%,%sess-rxpackets%,%sess-siptxbytes%,%sess-
siprxbytes%,%sess-miptxbytes%,%sess-miprxbytes%,%aaa-ttlreq%,%aaa-curreq%,%aaa-
ttlauthreq%,%aaa-curauthreq%,%aaa-ttlauthprobe%,%aaa-curauthprobe%,%aaa-ttlacctreq%,%aaa-
curacctreq%,%aaa-ttlauthsucc%,%aaa-ttlauthfail%,%aaa-ttlauthpurged%,%aaa-
ttlauthcancelled%,%aaa-ttlauthdmuchal%,%aaa-ttlradauthreq%,%aaa-curradauthreq%,%aaa-
ttlradauthreqretried%,%aaa-ttllclauthreq%,%aaa-curlclauthreq%,%aaa-ttlpseudoauthreq%,%aaa-
curpseudoauthreq%,%aaa-ttlauthnulluser%,%aaa-ttlacctsucc%,%aaa-ttlacctpurged%,%aaa-
ttlacctcancelled%,%aaa-ttlradacctreq%,%aaa-ttlradacctreqretried%,%disc-reason-summary%

card         cardsch
EMS,Card,%localdate%,%localtime%,%card%,%cpubusy%,%cpuidle%,%numproc%,%memused%,%memtotal%

port         portsch
EMS,Port,%localdate%,%localtime%,%card%,%port%,%rxbytes%,%txbytes%,%ucast_inpackets%,%ucas
t_outpackets%,%mcast_inpackets%,%mcast_outpackets%,%bcast_inpackets%,%bcast_outpackets%,%r
xpackets%,%txpackets%,%rxdiscbytes%,%rxdiscpackets%,%txdiscbytes%,%txdiscpackets%,%maxrate
%,%frag-rcvd%,%pkt-reassembled%,%frag-tokernel%
```

The values for the header fields are described in the table below:-

Header Field	Value	Meaning
<schema_type>	card	This is the identifier for the **Counter Block**.
<schema_name>	cardsch	The name or identifier of this schema. This is not used.
<schema_format>	EMS,Card,%localdate%,%localtime%,%card%,%cpubusy%	Specifies the content and format of the statistics. *EMS,Card* is used as the **identifier** to map the config data to the Performance data. The counters are specified in *%counter_name%*. i.e. Counters: localdate localtime card cpubusy

1.2.8.2 Performance data layout

Below are sample performance data records:

```
EMS,Card,20071023,084500,5,14.97,85.03,193,1352624,4194304

EMS,Port,20071023,084500,23,4,186335,0,567,0,1044,0,188,0,1799,0,0,2,0,0,100000000,0,0,0

EMS,PDSNSystem,20071023,124500,20739751,7339,11296885,45774569,8444003,22492492,15215091,2
798527,0,13716,11582,1932,4466,9250,0,9,0,0,1,1,2,8790008,2505,8785259,2243,11218,10546820
,2598887,7947933,7945928,1008,3516441,0,3516441,3515243,930,103,70,221,652,1720,2148,3442,
5359,1,2062335,590451,7025,1181,5556,189247,525293,544898,721575,579493,4273698,635982,219
124,233432,154278,125302,83033,55069,98787,4884,0,20071023,124504,20071023,084504,9643111,
13664242,158D 3H
37M,131000,25000,0,1000,180000852393421,41851331050141,206553138916,175132800655,169440524
513983,39507657739485,10560327879438,2343673310656,439368072,55,75949285,1,0,0,363418575,5
4,65250378,10612624,7379,78903,0,65586339,1,282959,2857,0,10360301,0,0,363204788,0,213731,
363414225,4312677,1=31201;2=77393;9=625624;11=6529671;14=15499;15=2;22=50742;23=37450;24=2
09133;25=1026423;27=173982;31=67744;33=5854699;34=83828;37=419;39=753219;40=1315598;41=290
662;43=194;47=3826;55=293206;58=8817;59=67;60=12087;61=51250;69=1006136;71=2201358;99=1;11
5=5488;126=199;142=85;146=1;161=29;170=1;186=1;
```

Using the **identifier** in the <schema_format> from the tables in *Bulk Statistics Configuration data*, the following is the mapping between the bulk statistics configuration data and the performance data:-

i.e.

From configuration data:

card cardsch
EMS,Card,%localdate%,%localtime%,%card%,%cpubusy%,%cpuidle%,%numproc%,%memused%,%memtotal%

From performance data:

EMS,Card,20071023,084500,5,14.97,85.03,193,1352624,4194304

Counter Block	**Counter**	**Value in counter**
Card	localdate	20071023
Card	localtime	084500
Card	card	6
Card	cpubusy	14.97
Card	cpuidle	85.03
Card	numproc	193
Card	memused	1352624
Card	memtotal	4194304

The number of counters in the schema format of the bulk statistics configuration file must be equal to the number of values available in the performance data.

Note:

Performance data can come from two different server types – FA (foreign agent) & HA (home agent). In order to generate reports per server type, they need to be separated into two directories and they will be differentiated through PDSN16000_Id (see section 10 regarding PDSN16000_Id).

1.2.8.3 File names

The file naming convention for Schema and Performance data are both configurable. The file naming convention below is the recommended convention based on sample data received from several customers:

- **Performance data:**

The recommended file naming convention for the performance data file is:-

bulkstatyyyymmddhhmmss.csv

- **Schema:**

There is no fixed file naming convention for the Schema file.

The recommended file naming convention for the configuration data is:-

bulkstat_configuration_yyyymmdd.txt

1.2.9 Architectural extensions

No external tools were used to parse the Nortel performance data.

2. Engine Rules and Configuration

The Nortel CDMA Gateway consists of the following parser modules to parse raw performance and configuration data.

- Nortel_MTX_Table
- Nortel_EVDO
- BSMMAP_PARSER
- Nortel_PDSN

2.1 Nortel_MTX_Table

The parser module is used to parse the configuration table data.

2.1.1 Rule Configuration

To evoke the Nortel_MTX_Table module the configuration option 'RULE_TYPE' must be set to ‘Nortel_MTX_Table’. The configuration options 'RULE_TYPE', 'RULE_DESC', 'INPUT_FILE_DESCRIPTION', 'INPUT_DIR_DEPTH', 'FILENAME_HEADER_FIELDS' and 'DIRECTORY_HEADER_FIELDS' are common to all productised gateways and are fully described in Gateways Framework User Guide; all other configuration options listed are specific to the Nortel_MTX_Table module.

Mandatory configuration entries:

- TABLE_NAME_DESCRIPTION: This is a list of table names to be parsed from the MTX Table config file.

```
TABLE_NAME_DESCRIPTION     => ['CDMACELL', 'NOISBSC'],
```

2.1.2 PIF naming convention

The PIF will be output with a name in the following format where '-#-' is used as a delimiter:

MTX3-#-SBSINV-#-ASCII_DATA-#-I.pif

<mtxid>-#-<table name>-#-ASCII_DATA-#-I.pif

2.2 Nortel_EVDO

This parser module parses raw data and configuration files into PIF format.

2.2.1 Rule Configuration

To evoke the Nortel_EVDO module the configuration option 'RULE_TYPE' must be set to 'Nortel_EVDO'. The configuration options 'RULE_TYPE', 'RULE_DESC', 'INPUT_FILE_DESCRIPTION', 'INPUT_DIR_DEPTH', 'FILENAME_HEADER_FIELDS' and 'DIRECTORY_HEADER_FIELDS' are common to all productised gateways and are fully described in Gateways Framework User Guide; all other configuration options listed are specific to the Nortel_EVDO module.

Mandatory configuration entries:

- FIELD_SEPARATOR: This is the Key to the field splitting process and controls the fields that are used internally. The field separator delimits the data.

  ```
  FIELD_SEPARATOR => '|',
  ```

- DATA_RECORD_SEPARATOR: This is a mandatory string that separates the header section and data section.

  ```
  DATA_RECORD_SEPARATOR => '^--------------------*$',
  ```

Optional components:

- HEADER_INFO_FOR_BLOCK_NAME: Uses a header info for data block name.

  ```
  HEADER_INFO_FOR_BLOCK_NAME => 'TYPE',
  ```

- HEADER_COUNTERS_TO_KEY_PIF_FILENAME: A list of header info to construct the output file name.

  ```
  HEADER_COUNTERS_TO_KEY_PIF_FILENAME => [ 'TYPE', 'TIME',
  'FILENAME' ],
  ```

- DEFAULT_NULL_VALUE: A value that is output when data is null.

  ```
  DEFAULT_NULL_VALUE => 'NULL',
  ```

- TIMESTAMP_FILE: The filename of the path and file name where the last time stamp for each element for all object type is stored. The time stamp is used for filtering records that has already been parsed and being the parsing from the last time stamp onwards. The filename will be concatenated with the date of the record data.

  ```
  TIMESTAMP_FILE => '.EVDO.timestamp',
  ```

- DATE_INFO_FIELDS: This is specify the data and time for the TIMESTAMP_FILE. The specific entry names are:
  - DATE: The date from the record data info that specifies the date.
  - TIME: The time from the record data info that specifies the time.

    ```
    DATE_INFO_FIELDS => {
        DATE => 'ENDDATE',
        TIME => 'ENDTIME',
    }
    ```

- OBJECT_TYPE_NAME_FOR_HEADER_INFO: The name for the header info for entity name.

  ```
  OBJECT_TYPE_NAME_FOR_HEADER_INFO => 'ENTITY',
  ```

- OBJECT_TYPE: A hash defining the entity name and output block name for each object type. The key for OBJECT_TYPE refers to the object type. In each object type key is a hash defining the Entity Name.

  An object type may have 1 or more Entity Name depending on the data field description.

Each Entity Eame consists of a hash that may define the OUTPUT_BLOCK_NAME, FIELD_DESCRIPTION, and ADD_DATA_FIELDS. All these are optional entries for each entity name hash.

- OUTPUT_BLOCK_NAME: The block name for the entity of the object type.
- FIELD_DESCRIPTION: The pattern to match the data field of the record to be defined with the entity name.
- ADD_DATA_FIELDS: A hash of new data fields to be created from within the records' header or data field. Each new data field is a key to a hash consisting of HEADER_FIELD_NAME or DATA_FIELD_NAME, FIELD_DESCRIPTION, FIELD_PREFIX, and FIELD_SUFFIX.
    - HEADER_FIELD_NAME: The header info field name
    - DATA_FIELD_NAME: The data info field name
    - FIELD_DESCRIPTION: The pattern to match the field value with a single sub pattern capture.
    - FIELD_PREFIX: A string to be prefixed to the new field value.
    - FIELD_SUFFIX: A string to be prefixed to the new field value.

A sample OBJECT_TYPE is defined below, where 'CCHStatsSectorCarrier' is the object type key; DOM_Sector is the Entity Name; IPADDRESS and SECTORELEMENTNAME are new data fields to add to the data record:

```
OBJECT_TYPE => {
      'CCHStatsSectorCarrier' => {
            DOM_Sector => {
                  OUTPUT_BLOCK_NAME => 'SectorCarrier',
                  ADD_DATA_FIELDS => {
                  IPADDRESS => {
                        HEADER_FIELD_NAME => 'NodeName',
                        FIELD_DESCRIPTION =>
                              'DOM_(\d+.\d+.\d+.\d+)',
                  },
                  SECTORELEMENTNAME => {
                        DATA_FIELD_NAME => 'sectorElementIndex',
                        FIELD_PREFIX => 'element',
                  },
            },
      },
},
```

- UPPERCASE_COUNTERS: When set to 'True', all counter names will be converted to uppercase. Set to 0 to disable.

```
UPPERCASE_COUNTERS => 'True',
```

- HEADER_DATA_RECORD_PROCESSING: This entry enables extra Perl code in the engine to do various manipulations to header and data records. This is the last process within the engine before the data is being output to the PIF files.

```
HEADER_DATA_RECORD_PROCESSING => sub {
   my ($$blkname_ref, $h_ref, $d_ref) = @_;

   # Header record processing
```

```
         if ($h_ref->{TIME} =~ /(\d{4})(\d{2})(\d{2})(\d{2})(\d{2})/) {
            my $day = "$1\-$2\-$3";
            my $time = "$4\:$5";
            &GenUtils::convert_date_format(\$day);

            # TIME will be output as a key PIF filename
            # as defined in HEADER_COUNTERS_TO_KEY_PIF_FILENAME
            # It is used by the UNPEGGER during post parsing
            $h_ref->{TIME} = "$day\_$time";
         }

         return 0; # Return 0 if successful
      },
```

2.2.2 PIF naming convention

The PIF will be output with a name in the following format where '-#-' is used as a delimiter:

```
<header_counters_to_key_pif_filename>-#-<value of block name>-#-
<counter>-#-I.pif
```

where <counter> will either be nonexistent (as in the case when the first file of <original filename> is processed) or it will be an integer (1,2,3,..) (as in the case where a file with same name has file already parsed and whose the resulting PIF still exists in the intermediate directory).

2.2.3 Nortel_EVDO_Hierarchy_Config

This configuration module is for the construction of the Nortel EVDO Element ID used in the Nortel_EVDO gateway, as well as in Post Parser configuration.

It contains a hash known as ENTITY_HASH which consists of the details for ENTITY name and element ID. The ENTITY_HASH is traversed recursively to construct the FDN and the element ID.

Below is an example of the ENTITY_HASH:

```
      my %ENTITY_HASH = (
          SYSTEM_ID => {
              '.*' => {
                  HEADER_INFO_FIELD => 'SYSTEM_ID',
              },
          },
          NETWORK_ID => {
              '.*' => {
                  HEADER_INFO_FIELD => 'NETWORK_ID',
              },
          },
          DO_RNC_ID => {
              '.*' => {
                  HEADER_INFO_FIELD => 'RNC_ID',
              },
          },
          DO_RNC => {
              FDN => [ 'SYSTEM_ID', 'NETWORK_ID', 'DO_RNC_ID' ],
          }
          DO_PDSN_ID => {
              '.*' => {
                  DATA_INFO_FIELD => 'pdsnNumberPdsn',
              },
          },
          DO_PDSN => {
              FDN => [ 'DO_RNC', 'DO_PDSN_ID' ],
          },
      )
```

The key for the ENTITY_HASH consists of Entity Name and Element ID.

When the key is an Entity Name, a hash consisting of an FDN entry is expected which is defined with a lists of Element ID.

```
          DO_PDSN => {
              FDN => [ 'DO_RNC', 'DO_PDSN_ID' ],
          },
```

When the key is an Element ID, a hash consisting of one or more Object Type entries are expected to retrieve data information from either the header or data record. Each Object Type entry is a hash consisting of the following:

- HEADER_INFO_FIELD or DATA_INFO_FIELD: Either the header or data info field from the record.
- HEADER_INFO_DESCRIPTION or DATA_INFO_DESCRIPTION: The pattern matching the header or data info consisting of a single sub pattern capture.

```
          DOM_ID => {
              '.*' => {
                  HEADER_INFO_FIELD => 'NodeName',
                  HEADER_INFO_DESCRIPTION => 'DOM_(.*)',
              },
```

2.2.4 CARD_INFO

Card info consists of a hash based on the hardware configuration table stated in the vendor documentation. It is a static lookup table to obtain the card slot ID and CPU ID.

The Nortel_EVDO_FDN_Config uses the CARD_INFO to lookup RNCCardID, DOMCardID, RNCPUID and DOMCPUID.

The Object type key for the Element ID hash consists of a hash entry, CARD_INFO_FIELD.

- CARD_INFO_FIELD: A hash entry for obtaining the element card ID or element CPU ID based on the subentries below.
    - HASH_REF: The hash reference in the CARD_INFO, either RNCCSlotHash, RNCCPUHash, DOMSlotHash, or DOMCPUHash.
    - DATA_ID: The data record field consisting the ID to refer the HASH_REF, either a slot number or entity CPU ID.
    - HASH_FIELD: The ID to return for the element ID, either RNCCardID, RNCCPUID, DOMCardID, or DOMCPUID.

The following is an example of the Element ID configuration in the ENTITY_HASH:

```
DOM_CPU_ID => {
    'CPUUtilization_2.2' => {
        CARD_INFO_FIELD => {
            HASH_REF   => 'DOMCPUHash',
            DATA_ID    => 'airEntCPUID',
            HASH_FIELD => 'DOMCPUID',
        },
    },
},
```

2.3 BSMMAP_PARSER

This is an additional rule in the Nortel CDMA parser to parse BSMmap files for look up purposes. This particular instance the CDMA_FREQ counter is inserted from these configuration files and used as the SECTOR_CARRIER_ID in the FDN.

2.3.1 Rule Configuration

To evoke the BSMMAP_PARSER module the configuration option 'RULE_TYPE' must be set to 'BSMMAP_PARSER'. The configuration options 'RULE_TYPE', 'RULE_DESC', 'INPUT_FILE_DESCRIPTION', 'INPUT_DIR_DEPTH', 'FILENAME_HEADER_FIELDS' and 'DIRECTORY_HEADER_FIELDS' are common to all productised gateways and are fully described in Gateways Framework User Guide; all other configuration options listed are specific to the BSMMAP_PARSER module.

Mandatory configuration entries:

- FIELD_SEPARATOR: This is the key to the field splitting process and controls the fields that are used internally. The field separator delimits the data.

```
FIELD_SEPARATOR => '|',
```

- DATA_FIELDS_DESCRIPTION: This is a hash of field names and regular expressions for the counter info fields.

```
DATA_FIELDS_DESCRIPTION => {
      CCLN              => '(CBS\d+)',
      CELLS             => '(Cells\d+)',
      MCBTS             => '(\w+1900BTS\d+)',
      MCBTSSUBSYSTEM    => '(\w+BTSSubsystem\d+)',
      ROOT              => '(Root\d+)',
      BTSCALLPROCESSING => '(BTSCallProcessing\d+)',
      ADVANCEDFA        => '(AdvancedFA\d+)',
      ADVANCEDSECTOR    => '(AdvancedSector\d+)',
      SECTOR_ID         => 'SectorId \= (\w+)',
      SECTOR_CELL_ID    => 'SectorCellId \= (\d+)',
      DCG               => '(DCG\d+)',
      SPLIT_MODE        => 'SplitMode \= (\w+)',
      SPLIT_MATE        => 'SplitMate \= (\w+)',
      CDMA_FREQ         => 'CDMA_FREQ \= (\d+)',
},
```

- BLOCK_NAME_DESCRIPTION: This is a hash of field names and regular expressions for the block names.

```
BLOCK_NAME_DESCRIPTION => {
        CDMA_FREQ              => 'CDMA_FREQ \= \d+',
        SPLITMODE_SPLITMATE    => 'SplitMate \= \w+',
        SECTORID_SECTORCELLID => 'SectorId \= \w+',
},
```

2.3.2 PIF Naming Convention

The output file will be the following format :

CDMA_FREQ-#-***MSC_NAME***-#-I.pif

Where ***MSC_NAME*** is the MSC, which this configuration file is for. The filename is intentionally generic so that when new configuration is parsed it will be overwritten.

2.4 Nortel_PDSN

This is an additional rule in the Nortel CDMA parser to parse PDSN files. Counter names are extracted from PDSN schema files and mapped with counter values from the performance data files.

2.4.1 Rule Configuration

To evoke the Nortel_PDSN module the configuration option 'RULE_TYPE' must be set to 'Nortel_PDSN'. The configuration options 'RULE_TYPE', 'RULE_DESC', 'INPUT_FILE_DESCRIPTION', 'INPUT_DIR_DEPTH', 'FILENAME_HEADER_FIELDS' and 'DIRECTORY_HEADER_FIELDS' are common to all productised gateways and are fully described in Gateways Framework User Guide; all other configuration options listed are specific to the Nortel_PDSN module.

Mandatory configuration entries:

- SCHEMA_DIR: a mandatory string representation of the directory path where the schema files are stored. The directory will be searched for schema files that match the SCHEMA_DESCRIPTION entry

  SCHEMA_DIR => '/gateways/nortel-pdsn/input_d/schema/'

- SCHEMA_DESCRIPTION: an array of file name regular expressions that match and identify the files as a schema file.

  SCHEMA_DESCRIPTION => ['^PDSN_Schema_\d+-\d+-\d+\.txt$']

- SCHEMA_HEADER_LEADTEXT: a string or regular expression indicating the start of the header data section in the schema file. If undefined, it is assumed that the first line of the file is the header data line.

  SCHEMA_HEADER_LEADTEXT => 'Schema for File'

- SCHEMA_HEADER_ROWS_TO_SKIP: an integer value indicating the number of header data lines to skip. If undefined, the default value is zero (0).

  SCHEMA_HEADER_ROWS_TO_SKIP => 2

- SCHEMA_COLUMN_SEPARATOR: a string value indicating the column separator. If undefined, the default value is a single white space ( ). It is used to distinguish the different elements in a schema record row.

  SCHEMA_COLUMN_SEPARATOR => ' '

- SCHEMA_COUNTER_MARKER: a string value indicating the start and end of a counter name. If undefined, the default value is a percent character (%).

SCHEMA_COUNTER_MARKER => '%'

- SCHEMA_COUNTER_DELIMITER: a string value separating individual counters in the schema record. If undefined, the default value is a coma (,).

  SCHEMA_COUNTER_DELIMITER => ','

- DATA_HEADER_LEADTEXT: a string or regular expression indicating the start of the header data section in the performance data file. If undefined, it is assumed that the first line of the file is the header data line.

  DATA_HEADER_LEADTEXT => 'Format 5.0'

- DATA_HEADER_ROWS_TO_SKIP: an integer value indicating the number of header data lines to skip. If undefined, the default value is zero (0).

  DATA_HEADER_ROWS_TO_SKIP => 0

- DATA_COUNTER_DELIMITER: a string value separating individual counters in the performance data record. If undefined, the default value is a coma (,).

  DATA_COUNTER_DELIMITER => ','

- DATA_MAX_SCHEMA_MATCH_LEVELS: an integer value indicating the maximum number of counters to be used when matching a data record with the schema mapping. The parser will start with the maximum number of counters and work backwards until a matching schema map is found or the minimum counter (1) match is reached. If undefined, the default value is 2.

  DATA_MAX_SCHEMA_MATCH_LEVELS => 2

2.4.2 PIF Naming Convention

Each raw file may produce multiple PIF files. Each PIF file contains only one specific block type.

The output file name will be in the following convention:

FILE_NAME-#-PARENT_BLOCK-#-BLOCK_NAME-#-I.pif

Where

FILE_NAME is the file name of the raw file

PARENT_BLOCK is the name of the parent block mapped from the Schema file

BLOCK_NAME is the name of the blocks contained within the PIF file extracted from the record data

3. Post Parser Rules and Configuration

No customized post parser rules are provided with Nortel CDMA Gateway.

4. Virtuo DA Configuration

Virtuo DA toolkit is used to obtain the configuration data for MTX Table and BSMmap, and MTX password rotation functionality. Please refer to the README in the Virtuo DA toolkit that comes with the Nortel CDMA Gateway for installation and usage.

4.1 MTX Table

The Nortel EVDO, MTX13 and RAN require data extraction from MTX table especially for the FDN construction. Data from MTX table is normally incorporated into the performance data by the vendor gateway using the post parser rule. MTX Table parser module is used to do the data extraction from the MTX Table file.

4.1.1 DA Configuration

User needs to update the following configurations accordingly in order for the scripts to run.

The MTX Table configuration is like below:

```
set MTXIP             ???.???.???.???
set MTXPORT           23
set MTXUSER           username
set MTXPSWD           password
set MTXNAME           MTX01
set LOCAL_MTXCFGDIR   .
set MTXTABLES         "C7RTESET C7LKSET CLLI CDMACELL TRKMEM PWRCTRL
HOPARMS VCHINV CELLULAR SBSINV"
set GWDESTDIR         "/<gways_path>/in_dir"

#
# Path for executables
#
set TELNET  /bin/telnet

#
# Timeouts (sec)
#
set GETTABLETIMEOUT   120

#
# Log
#
set LOGFILENAME  /<log_path>/mtxtablesdist.log
set LOGLEVEL     5

#
# Password rotation
#
set PWDLOGFILENAME  mtxpwd.log
set PERL5           /opt/perl-5.6.1/bin/perl
set GENPWD          genpwd.pl
```

Descriptions:

Parameter	Descriptions
MTXIP	IP address of the MTX
MTXPORT	Port ID of the MTX

MTXUSER	MTX username
MTXPSWD	MTX login password
MTXNAME	MTX name
LOCAL_MTXCFGDIR	Local directory for MTX configuration
MTXTABLES	List of required MTX tables
GWDESTDIR	Gateway destination directory (IN_DIR)
TELNET	Telnet path
GETTABLETIMEOUT	Timeout to get the table
LOGFILENAME	Filename of the log
LOGLEVEL	Log level of the log
PWDLOGFILENAME	MTX password log file name
PERL5	Perl directory
GENPWD	Generate password script

4.2 BSMmap

The Nortel RAN requires data extraction from BSMmap especially for the FDN construction. Data from BSMmap is normally incorporated into the performance data by the vendor gateway using the post parser rule. BSMmap parser module is used to do the data extraction from the BSMmap file.

4.2.1 DA Configuration

The BSMmap configuration is like below:

```
set REMOTEHOST  ???.??.???.??
set BSMUSER     username
set BSMPSWD     password
set MTXNAME     "MTX"
set BSMID       "BSM"
set BSMIP       "???.??.???.??"
set BSMOMDIR    "DIR"
set LOCALDIR    "."
set FLEXPMHOST  ???.??.???.??
set FLEXUSER    username
set FLEXPSWD    password
set FLEXDIR     "/<flex_path>/flexpm"
set NDAYS       1
set GWDESTDIR   "/<gways_path>/in_dir"


#
# Path for executables
#
set TELNET          /bin/telnet
set REMOTEDATE      /usr/bin/date
set REMOTECLIAPP    /<remote_path>/cliapp
set REMOTERM        /usr/bin/rm


#
# Timeouts
#
set INITTIMEOUT      180
set RDATETIMEOUT     60
set GENERATETIMEOUT  2

#
# Log
#
set LOGFILENAME  bsmmapdist.log
set LOGLEVEL     5
```

Descriptions:

Parameter	Descriptions
REMOTEHOST	Remote host IP address
BSMUSER	BSM username
BSMPSWD	BSM password
MTXNAME	MTX name
BSMID	BSM ID
BSMIP	BSM IP address
BSMOMDIR	BSM ON directory
LOCALDIR	Local directory for BSM configuration
FLEXPMHOST	Flexpm host IP address
FLEXUSER	Flexpm username
FLEXPSWD	Flexpm password
FLEXDIR	Flexpm directory
NDAYS	Number of days to keep the file
GWDESTDIR	Gateway input directory (IN_DIR)
TELNET	Telnet path
REMOTEDATE	Date path
REMOTECLIAPP	CLIAPP path
REMOTERM	Term path
INITTIMEOUT	Initialisation timeout
RDATETIMEOUT	Remote date timeout
GENERATETIMEOUT	Generate timeout

LOGFILENAME	File name of the log
LOGLEVEL	Log level of the log

4.3 MTX Password Rotation

A part from Nortel Centennial requirement, it is required to have a script to do password changes of the MTX box for every couple of month for security reason. Thus this Virtuo DA has a module/script to do such implementation.

5. Configuration Support

Gateway Configurations listed below are available for the respective Tech Pack releases:

- Nortel MTX
- Nortel EVDO
- Nortel NBSS
- Nortel PDSN