



QXDM Software Users Guide

80-V1241-21 Rev. A

November 21, 2001

Restricted Distribution: This document contains critical information about QUALCOMM products and may not be distributed to anyone that is not an employee of QUALCOMM without the approval of Configuration Management.

**QUALCOMM Incorporated
5775 Morehouse Dr.
San Diego, CA 92121-1714
U.S.A.**

QUALCOMM Incorporated
5775 Morehouse Dr.
San Diego, CA 92121-1714
U.S.A.

Copyright © 2000, 2001 QUALCOMM Incorporated. All rights reserved.

All data and information contained in or disclosed by this document is confidential and proprietary information of QUALCOMM Incorporated, and all rights therein are expressly reserved. By accepting this material, the recipient agrees that this material and the information contained therein is held in confidence and in trust and will not be used, copied, reproduced in whole or in part, nor its contents revealed in any manner to others without the express written permission of QUALCOMM Incorporated.

Export of this technology may be controlled by the United States Government. Diversion contrary to U.S. law prohibited.

Restricted Distribution: This document contains critical information about QUALCOMM products and may not be distributed to anyone that is not an employee of QUALCOMM without the approval of Configuration Management.

QUALCOMM is a registered trademark and registered service mark of QUALCOMM Incorporated. Other product and brand names may be trademarks or registered trademarks of their respective owners.

Information in this document is preliminary and subject to change and does not represent a commitment on the part of QUALCOMM Incorporated.

QXDM Software Users Guide

80-V1241-21 Rev. A

November 21, 2001

Contents

1 Introduction.....	1-1
1.1 Purpose	1-1
1.2 Scope and intended audience.....	1-1
1.3 Conventions.....	1-1
1.4 Revision history.....	1-2
1.5 References	1-4
1.6 Technical assistance	1-4
1.7 Acronyms	1-4
2 Installation.....	2-1
2.1 Required hardware and software	2-1
2.2 Installing QXDM.....	2-1
2.3 Installing the license file.....	2-2
2.4 Physical connectivity.....	2-2
3 Communication Parameters	3-1
4 Display Overview	4-1
4.1 Options	4-1
4.2 Status bar	4-5
4.3 Multiple views.....	4-7
4.4 Additional views.....	4-11
4.4.1 Command Output view.....	4-11
4.4.2 Script Help view	4-11
4.4.3 Message view and Autoscroll button.....	4-11
4.4.4 Property view.....	4-11
4.4.5 Memory viewer.....	4-12
5 Logging	5-1
6 Properties.....	6-1
6.1 Property definition.....	6-3

6.2 Property packets.....	6-3
6.3 Absolute address on the phone	6-4
6.4 Properties with arrays	6-5
6.5 Structures in properties	6-6
6.6 Property support tools	6-7
6.6.1 makeProp tool	6-8
6.6.2 makeHTML tool.....	6-8
7 Events	7-1
7.1 Event information	7-1
7.2 Event display.....	7-1
7.2.1 Scrolling display.....	7-2
7.2.2 Table display	7-3
7.3 Event options	7-5
7.4 Event filtering	7-6
7.4.1 Basic event filtering	7-7
7.4.2 Advanced event filtering	7-7
7.5 Custom event display	7-8
8 Property Editor	8-1
8.1 Built-in Property database.....	8-1
8.2 User Property database	8-1
8.3 Property Editor User Interface	8-2
8.3.1 Splitter Window Overview of the Property Editor.....	8-2
8.3.2 Category Display of the Property Editor.....	8-3
8.3.3 Property Display of the Property Editor.....	8-4
8.4 Working with properties	8-5
8.4.1 Inserting a new property.....	8-5
8.4.2 Renaming a property	8-6
8.4.3 Deleting a property.....	8-6
8.4.4 Modifying category selections	8-6
8.4.5 Modifying property field structure.....	8-6
8.5 Working with categories	8-6
8.5.1 Adding a new category.....	8-6
8.5.2 Renaming a category.....	8-7
8.5.3 Deleting a category	8-7
8.6 Working with property fields.....	8-7
8.6.1 Adding a new property field	8-8
8.6.2 Modifying property field contents.....	8-8
8.6.3 Deleting a property field	8-8
8.7 Saving the database.....	8-8
8.8 Conflicts.....	8-9
8.8.1 Resolving conflicts.....	8-10

9 Methods.....	9-1
10 DWARF Server	10-1
10.1 Installation	10-1
10.2 Build process	10-2
10.3 Access process.....	10-3
10.4 Configuring build machines in QXDM.....	10-4
11 Script Interface	11-1
12 QXDM Command-prompt Functions.....	12-1
13 Temporal Analyzer	13-1
13.1 Temporal Analyzer display	13-1
13.2 Pilot Sets display	13-2
13.3 Transmit/Receive (Tx/Rx) Adjusted Power display.....	13-3
13.4 Frame Error Rate (FER) display.....	13-4
13.5 Frame Types (old Tx/Rx Vocoder Rate) display.....	13-5
13.6 RLP Throughput display	13-6
13.7 Temporal Analyzer Configuration	13-7
14 Application Screens	14-1
14.1 Application screens	14-1
14.1.1 Note 1	14-7
14.1.2 Note 2	14-7
14.1.3 Note 3	14-7
14.1.4 Note 4	14-8
14.1.5 Note 5	14-9
14.1.6 Note 6	14-9
14.1.7 Note 7	14-9
14.1.8 Note 8	14-9
14.1.9 Note 9	14-9
14.1.10 Note 10	14-9
14.1.11 Note 11	14-10
14.1.12 Note 12	14-12
14.1.13 Note 13	14-12
14.1.14 Note 14	14-12
14.1.15 Note 15	14-12
14.1.16 Note 16	14-12
15 Troubleshooting	15-1

Figures

Figure 2–1 Physical connectivity	2-2
Figure 3–1 Options → Communications → Configuration tab	3-1
Figure 3–2 Options → Communications → Timeout Settings tab	3-2
Figure 3–3 Options → Communications → Statistics tab	3-3
Figure 4–1 Options → Dip Switch dialog	4-1
Figure 4–2 Options → Settings → Messages dialog	4-2
Figure 4–3 Options → Settings → Preferences dialog	4-3
Figure 4–4 Options → SIA Remote Control dialog.....	4-4
Figure 4–5 QXDM status bar.....	4-5
Figure 4–6 Splitter view bar	4-7
Figure 4–7 Example of one splitter screen.....	4-8
Figure 4–8 Example of two splitter screens.....	4-9
Figure 4–9 Example of three splitter screens.....	4-10
Figure 4–10 Memory viewer	4-12
Figure 5–1 Log Mask Configuration dialog	5-1
Figure 6–1 Command line access to properties	6-1
Figure 7–1 Scrolling event display	7-2
Figure 7–2 Table event display	7-3
Figure 7–3 Options → Event Options.....	7-5
Figure 7–4 Event filtering.....	7-6
Figure 8–1 Property Editor window	8-2
Figure 8–2 Category display	8-3
Figure 8–3 Property display	8-4
Figure 8–4 Select Property dialog.....	8-5
Figure 8–5 Field Edit dialog	8-7
Figure 8–6 Property Editor conflict warning	8-9
Figure 8–7 Conflict List dialog.....	8-10
Figure 8–8 Resolving conflicting property IDs	8-10
Figure 8–9 Resolving conflicting property names.....	8-11
Figure 10–1 Access process.....	10-3
Figure 10–2 Path to the ELF file.....	10-3
Figure 10–3 No Symbol Data Base Loaded message	10-4
Figure 10–4 Configure build machines.....	10-4
Figure 13–1 Temporal Analyzer display	13-1
Figure 13–2 Pilot Sets display	13-2
Figure 13–3 Transmit/Receive (Tx/Rx) Power display	13-3
Figure 13–4 Frame Error Rate (FER) display.....	13-4
Figure 13–5 Frame Types (old Tx/Rx Vocoder Rate) display.....	13-5
Figure 13–6 RLP Throughput display	13-6
Figure 13–7 Temporal Analyzer → Configuration.....	13-7
Figure 14–1 Register Peek/Poke Viewer	14-10
Figure 14–2 Mask dialog	14-11

Tables

Table 1-1 Revision history 1-2

Table 1-2 Reference documents 1-4

Table 3-1 Timeout settings 3-2

Table 6-1 Property file information..... 6-3

Table 14-1 Properties and packets..... 14-1

Table 14-2 Required syntax for send_is801_message 14-8

1 Introduction

1.1 Purpose

The QUALCOMM Extensible Diagnostic Monitor (QXDM) provides a diagnostic client for dual-mode subscriber station (DMSS) software. Its interface is based on the mobile diagnostic monitor (MDM) and the CDMA air interface tester (CAIT). It was developed to provide a rapid prototyping platform for new diagnostic clients and diagnostic protocol packets. QXDM provides a graphical user interface (GUI) that displays data retrieved from DMSS.

1.2 Scope and intended audience

This user guide is intended for users of QXDM who need to know how to install, use, and understand the information provided by QXDM. Many features in QXDM are documented elsewhere. Applicable documents are referenced in Table 1-2, however, most features are documented in the Compatibility Notes for the software release in which the feature was added. The following is an overview of what is covered in this guide:

- Installation
- Physical connectivity
- Communication parameters
- QXDM screens

QXDM also provides some advanced features designed for development:

- DWARF server
- Perl script interface

1.3 Conventions

Function declarations, function names, type declarations, and code samples appear in a different font. For example: `#include`

Code variables appear in angle brackets. For example: `<number>`

Shading indicates content that has been added or changed in this revision of the document.

1.4 Revision history

The revision history for this document is shown in Table 1-1.

Table 1-1 Revision history

Document number	Date	Description
80-V1241-1 Rev. –	Mar 2000	Initial release
80-V1241-3 Rev. –	Apr 2000	Added chapters: application screens, troubleshooting. Added status bar description. Note: There was no 80-V1241-2. The number was skipped intentionally in accordance with internal administrative tracking practices.
80-V1241-4 Rev. –	Apr 2000	Added temporal analyzer information. Updated application screens chapter.
80-V1241-9 Rev. –	Jun 2000	Added GPS feature. Added sections and screens: RLP throughput display, quick paging channel, RLP3 statistics, NV items. Updated chapters: installation, logging and events, application screens. Note: There was no 80-V1241-5 through 80-V1241-8. The numbers were skipped intentionally in accordance with internal administrative tracking practices.
80-V1241-14 Rev. –	Aug 2000	Added sections and screens: IS-2000 supplemental channel mux parameters, retrievable parameters, streaming configuration, simple test data services, gpsOne. Updated chapters: overview of screens, application screens. Updated screens: three splitter screen, NV items. Note: There was no 80-V1241-10 through 80-V1241-13. The numbers were skipped intentionally in accordance with internal administrative tracking practices.
80-V1241-15 Rev. –	Sep 2000	Added sections and screens: pilot sets display, temporal analyzer configuration, log priorities, GPS statistics, factory test mode. Updated properties chapter. Updated sections and/or screens: fast forward power control, IS-2000 supplemental channel mux parameters, retrievable parameters, simple test data services, gpsOne. Updated scope and intended audience, reference documents. Removed streaming configuration section.
80-V1241-17 Rev. –	Nov 2000	Renamed overview of screens chapter to display overview chapter. Added sections to the chapter: options (with screens), property views. Added sections and screens: full test data services, events, Bluetooth™ logs. Updated logging and events chapter and screen. Updated sections: property support tools, gpsOne, GPS statistics. Updated screen: simple test data services. Updated reference documents. Note: There was no 80-V1241-16. The number was skipped intentionally in accordance with internal administrative tracking practices.
80-V1241-18 Rev. –	Dec 2000	Added gpsOne reference document. Updated sections and/or screen: gpsOne, GPS statistics. Updated reference documents.
80-V1241-19 Rev. –	Feb 2001	Added property editor chapter. Updated properties chapter. Updated reference documents.

Document number	Date	Description
80-V1241-20 Rev. –	Mar 2001	<p>Added chapters: Events, QXDM command prompt functions.</p> <p>Updated Perl script interface chapter.</p> <p>Updated screens: mux traffic channel statistics, mux traffic channel and secondary statistics, quick paging channel statistics, fast forward power control logs, RLP3 statistics log, NV items, IS-2000 SCH mux parameters, retrievable parameters, BT logs, Bluetooth™ logs.</p>
80-V1241-21 X1	Jun 2001	<p>Updated Sections 4.1-Options, 4.4.3-Message view and autoscroll button, and 7.2.1-Scrolling display.</p> <p>Updated screens: RLP3 Statistics Log, IS-2000 SCH MUX Parameters, Bluetooth™ Logs, Full Test Data Services, gpsOne, Simple Test Data Services.</p> <p>Added screens Active Set Information and the following HDR screens: Air Link Summary, Fingers Data, Forward Link Statistics, GEN TA, Reverse Link Statistics, RLP Statistics, RX Statistics, SEARCH Status, Status, Temporal Analyzer, and the following WCDMA screens: AGC, Block Error Rate, Downlink TM Channel Parameters, Layer 1 State, Layer 4 Connection Mgmt, MAC Channel Mapping, MAC Parameters, Mobility Management, Physical Channel, RLC DL UM Channel Parameters, RLC UL UM Channel Parameters, RRC Status, Temporal Analyzer, TFCS Downlink, Transport Channels, Uplink TM Channel Parameters.</p> <p>Note: The document number for this revision has been changed to adhere to current document numbering standards.</p>
80-V1241-21 X2	Sep 2001	<p>Rewrote chapter 11 – Script Interface.</p> <p>Updated Annotate Logfile documentation, section 4.1. Updated ALIEM documentation, section 4.2.</p> <p>Updated Chapter 14 screens: Full Test Data Service, Simple Test Data Service.</p> <p>Removed Chapter 14 Sections: IS-2000 SCH MUX Parameters, WCDMA L4 Connection Management.</p> <p>Added Chapter 14 Sections: Mux Statistics, WCDMA CS and PS Connection Management, WCDMA NAS Error.</p>
80-V1241-21 Rev. A	Nov 2001	<p>Rewrote chapter 14 – Application Screens.</p> <p>Updated Property Support Tools documentation, section 6.6 and 6.6.1.</p> <p>Added documentation to Property Database sections 8.1 and 8.2.</p> <p>Updated document version numbers and dates in Section 1.5.</p> <p>The revision numbering system has been changed for internal tracking purposes only.</p>

1

2

1.5 References

Reference documents, which may include QUALCOMM, standards, and resource documents, are listed in Table 1-2.

Table 1-2 Reference documents

Ref.	Document		
QUALCOMM			
1	CDMA DMSS Serial Data Interface Control Document	80-V1294-1 X9	Sep 2001
2	Factory Test Mode Application Note	CL93-V1974-1 X3	Mar 2001
3	gpsOne™ Position Determination Messaging and Parameters	80-V0726-1 X6	Jan 2001
4	Serial Interface Control Document for W-CDMA	80-V2708-1 X2	Oct 2001

1.6 Technical assistance

For assistance or clarification on information in this guide, email QUALCOMM CDMA Technologies at asicapps@qualcomm.com.

1.7 Acronyms

The following acronyms are used throughout this guide:

CAIT	CDMA air interface tester
CDMA	Code division multiple access
DLL	Dynamic link library
DMSS	Dual-mode subscriber station
ELF	Executable and linking format
ESN	Electronic serial number
FER	Frame error rate
FFA	Form-fit accurate
F-FCH	Forward fundamental channel
F-SCCH	Forward supplemental code channel
F-SCH	Forward supplemental channel
FTM	Factory test mode
GPS	Global positioning system

GUI	Graphical user interface
HTML	Hypertext markup language
MDM	Mobile diagnostic monitor
MSM	Mobile station modem
NV	Nonvolatile memory
PDE	Position determination entity
PN	Pseudorandom noise
QXDM	QUALCOMM extensible diagnostic monitor
R-DCCH	Reverse dedicated control channel
R-FCH	Reverse fundamental channel
RLP	Radio link protocol/processor
R-SCH	Reverse supplemental channel
SIA	Sensor interface application
SILK	Structure Iteration Language toolkit
SURF	Subscriber unit reference platform

1
2

1

2

2 Installation

2.1 Required hardware and software

QXDM is designed to be installed and run on a workstation running Microsoft® Windows NT® 4.0 or Windows 98®.

NOTE An installation of Microsoft Internet Explorer 5.0 is required to run QXDM.

QXDM can be run on a Microsoft Windows 95® workstation but requires an installation of DCOM for Windows 95. Both Internet Explorer® Version 5.0 and DCOM for Windows 95 are available on the QXDM installer CD. A license file is also required, which enables QXDM to run for a specified period of time.

2.2 Installing QXDM

The QXDM software is provided either over the network or on CD-ROM.

This installer loads the user release of QXDM, which includes the QXDM client components (application binaries, DLLs, HTML files, property files, and so on) and the server components (Build ID and DWARF servers) on the user's machine.

A variant of this installer is the QXDM server installer, which installs only the QXDM server components on a designated build server machine.

The following procedure explains the user release installer only.

To install QXDM:

1. From the Windows desktop, select Start→Run→Setup.exe.
2. Click **OK**.
3. By default, QXDM will be installed on your machine at c:\Program Files\qualcomm\qxdm.

The installer will allow you to change the installation directory to one of your choosing.

The installer creates a QXDM folder in your Start menu. QXDM can be run from your Start menu by selecting Start → Programs → QXDM → QXDM.

2.3 Installing the license file

A license file is provided which enables QXDM to run for a specified period of time. The filename is `license.txt` and must be placed in the `bin` directory where QXDM was installed (for example, `C:\Program Files\qualcomm\qxdm\bin\license.txt`). The license file is a text file that can be viewed using any editor such as NotePad. It contains a readable start date and end date followed by two or more lines of encrypted data. To illustrate, the following license file enables QXDM to run from February 21, 2000 to March 22, 2000.

File: `license.txt`

Start Date: 2000 02 21

End Date: 2000 03 22

99 61 55 f5 19 33 ec 1d 15 eb 3d a2 76 e6 a8 40

da c0 b7 ea c5 67 04 3a 0b d6 0d d2 87 dc cd 40

Making changes to `license.txt` will render it invalid. If you need a new license file, contact QUALCOMM Technical Support by email at asicapps@qualcomm.com.

2.4 Physical connectivity

QXDM connects to a phone or SURF using a serial cable to a COM port on your PC, as depicted in Figure 2–1.

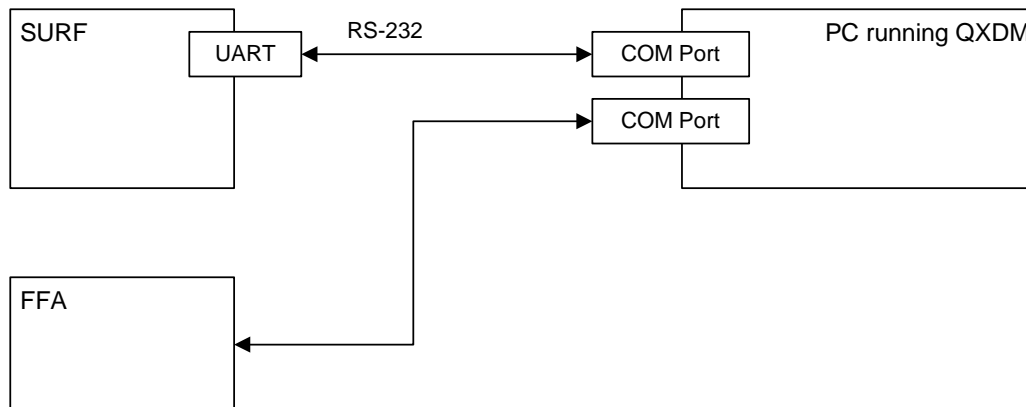


Figure 2–1 Physical connectivity

3 Communication Parameters

QXDM communication parameters are configured using the Options→Communications dialog. The Communications dialog allows you to configure the COM port that will be used by QXDM, as well as the baud rate and timeout settings. Figure 3–1 illustrates the configuration.

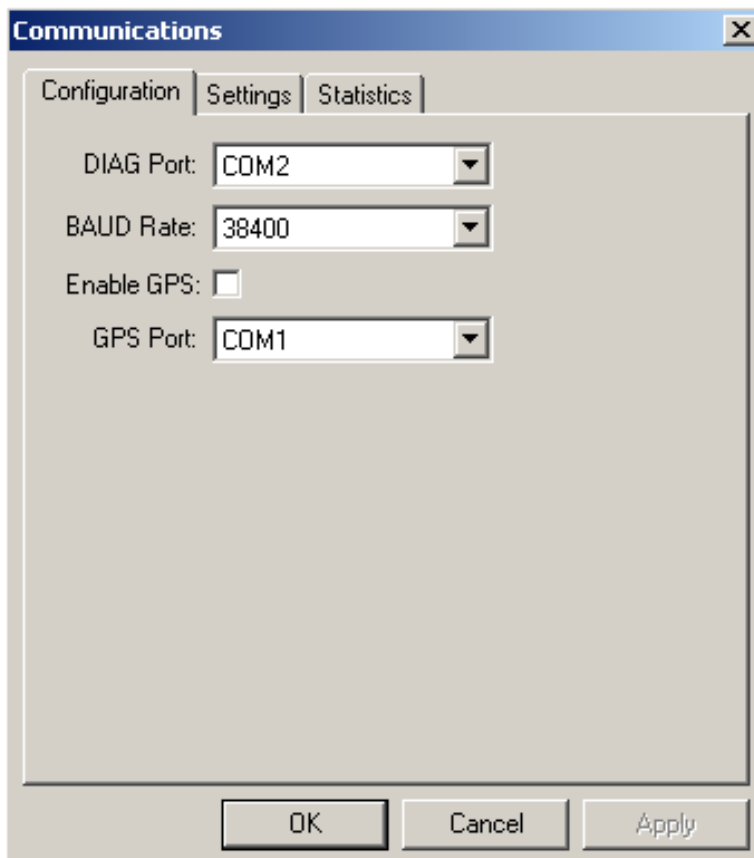


Figure 3–1 Options → Communications → Configuration tab

To use a connected GPS Receiver, select which COM port your GPS receiver is connected to, then select the Enable GPS check box. QXDM will start collecting GPS information from the GPS receiver.

The timeout settings dialog is accessible via the Options → Communications → Settings tab. Figure 3-2 illustrates the dialog.

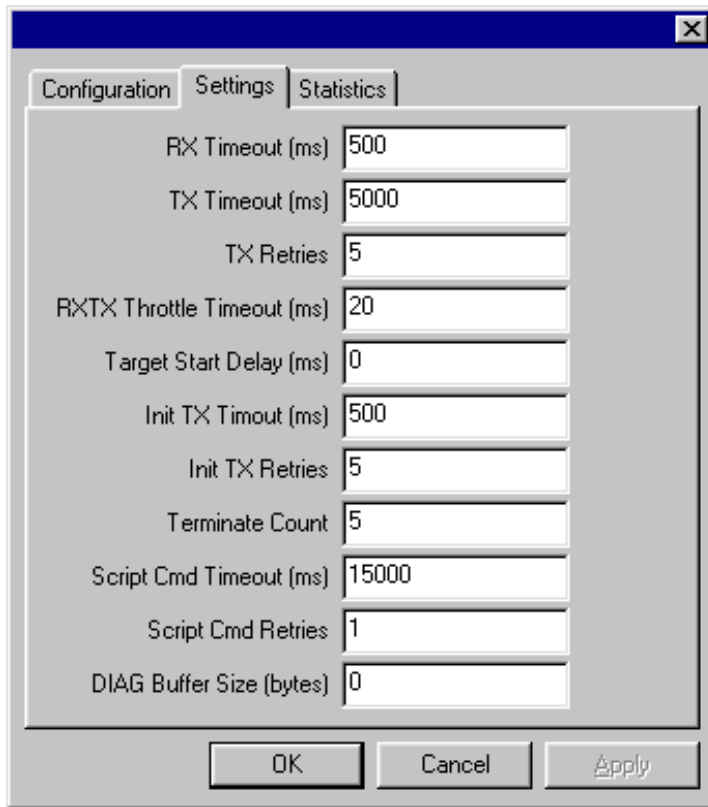


Figure 3-2 Options → Communications → Timeout Settings tab

Table 3-1 describes the timeout settings.

Table 3-1 Timeout settings

Timeout setting	Description
RX Timeout (ms)	Time QXDM waits without receiving a message before determining that the phone is disconnected.
TX Timeout (ms)	Time QXDM waits for a reply from the phone.
TX Retries	Number of times to retry waiting for a reply from the phone.
RXTX Throttle Timeout (ms)	Time between sends.
Target Start Delay (ms)	Time to wait after establishing a connection with the phone before sending the first request.
Init TX Timeout (ms)	Time to wait for a reply from the phone when reconnecting.
Init TX Retries	Number of times to retry reconnecting to the phone.
Terminate Count	Number of receive timeouts before determining that the phone is disconnected.
Script Cmd Timeout (ms)	Time to wait before timing out for script commands (command line).

Timeout setting	Description
Script Cmd Retries	Number of retries before giving up sending script commands (command line).
DIAG Buffer Size (bytes)	Unused.

The statistics display is accessible via the Options → Communications → Statistics tab. Figure 3–3 illustrates the display.

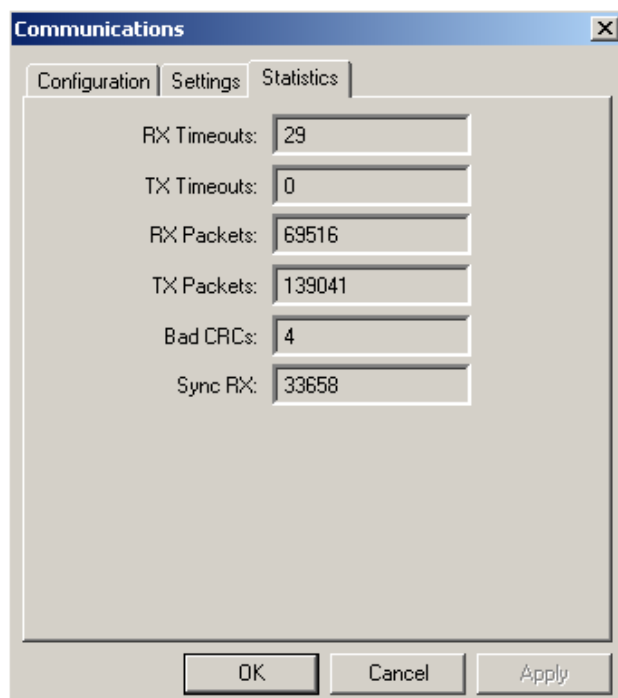


Figure 3–3 Options → Communications → Statistics tab

1

2

4 Display Overview

QXDM uses a single-document interface with multiple views.

4.1 Options

Communications

See Chapter 3 for more information.

Dip switches

The Dip Switch dialog allows you to configure supported dip switch settings, as shown in Figure 4–1.

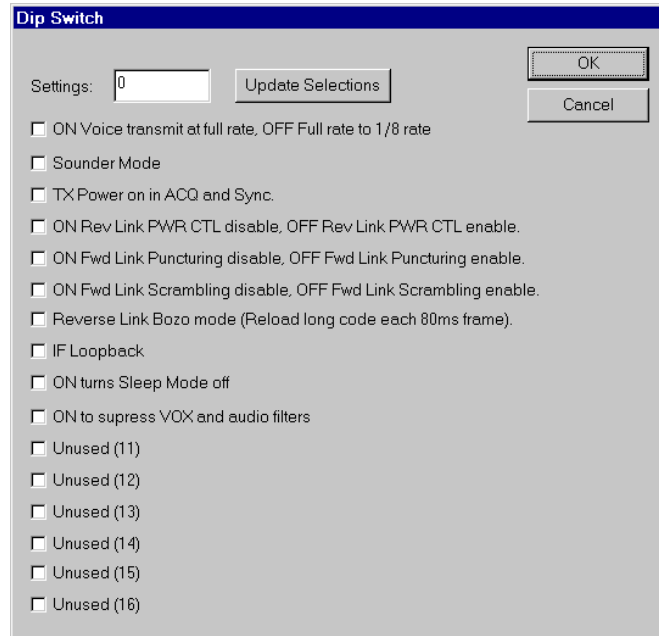


Figure 4–1 Options → Dip Switch dialog

Settings

The Settings display, shown in Figure 4–2, allows you to control which messages are displayed to the Messages <F3> screen, configure build machine settings, and set various preferences. See Section 10.4 for information on configuring build machines.



Figure 4–2 Options → Settings → Messages dialog

Settings → Preferences

This screen allows you to set various personal preferences. You can override the default path from which to run scripts (the default is C:\Program Files\Qualcomm\QXDM\bin). It is also possible to enter a path directly in the command line box. You can also establish a limit on how many lines are displayed on the Messages <F3> screen.

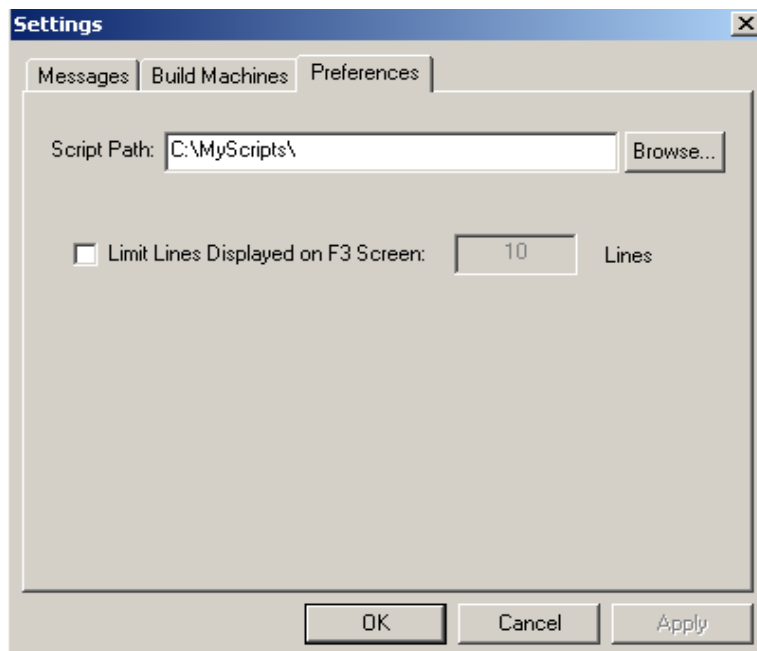


Figure 4–3 Options → Settings → Preferences dialog

Licensing

The Licensing command allows you to view your current QXDM license status.

Logging options

This dialog allows you to set the log mask. See Chapter 5 for more information.

Event options

The Event Options command allows you to configure the default settings used to display events. See Section 7.3 for more information.

Edit annotation list

The Edit Annotation List command allows you to select predefined comments to annotate log messages.

Annotate log file

Use this command to insert annotations to the log file. This command is enabled only during logging.

SIA remote control

The SIA (Sensor Interface Application) Remote Control dialog shown in Figure 4–4, allows you to start a remote session with a Position Determination Entity (PDE).

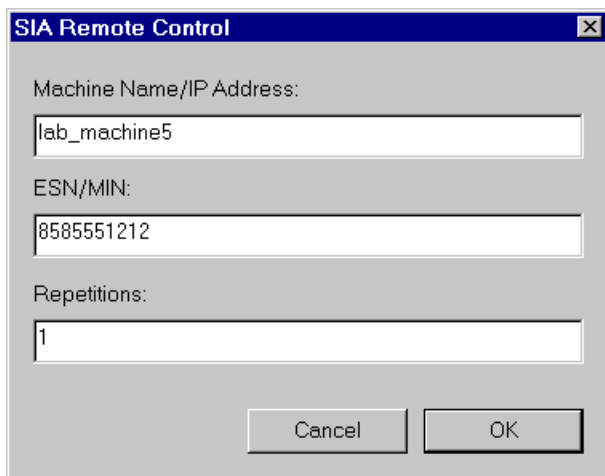


Figure 4–4 Options → SIA Remote Control dialog

Logging

The Logging command enables or disables (toggles) logging to a file.

Event reporting

The Event Reporting command enables or disables (toggles) display of events.

Flush F3 Screen

This command erases all messages from the screen.

4.2 Status bar

The status bar, shown in Figure 4–5, contains status information about the current session, as well as a command interface.



Figure 4–5 QXDM status bar

Command Box

You can type commands into the command box. Most DM script commands are supported. DM and Perl scripts can be started from the command box by typing **run script_file_name**, where `script_file_name` is the name of the DM script (.scr extension) or a Perl script (.pl extension). When launching a script using the run command, it may be necessary to specify the fully qualified path name of the file. Launching Perl scripts from the command box requires that Perl be installed and associated with the .pl file extension. A Command Output screen displays the output, and a Script Help page is available to help with DM script commands. See Section 4.4 for information on how to display the Command Output and Script Help screens.

Mode

This displays Streaming when a phone with a streaming build is connected, or Query when a phone with a nonstreaming build is connected.

RX rate

The rate at which data is received from the phone. It is displayed in BAUD, bits per second, or bytes per second depending on the option selected on the Options → Settings → Messages dialog (see Figure 4–2).

TX rate

The rate at which data is sent to the phone. It is displayed in BAUD, bits per second, or bytes per second depending on the option selected on the Options → Settings → Messages dialog (see Figure 4–2).

Total msgs

Total number of messages reported by the phone.

Dropped msgs

Total number of dropped messages reported by the phone.

ALIEM

QXDM provides a TCP Socket interface for the WCDMA AL1 Emulator (ALIEM) to send Diag commands to the connected SURF/Mobile through QXDM and receive the response. This Socket interface can also be used by clients other than the ALIEM. On startup, QXDM sets up a TCP listener on address 5999. The Status Bar in the bottom displays the state of the QXDM-ALIEM listener. Initially, the display is ALIEM: Ready. When a client connects to the socket, the display changes to ALIEM: Connected. Clients send Diag request packets (as explained in the DMSS ICD) through the Socket. The packet format is as follows:

```
struct ALIEMPacket
{
    uint16 len;
    uint8 data [min(len, 1620)];
}
```

QXDM passes the request packet to the SURF/Mobile and returns the response packet through the socket. The packet format is the same as the one for the request packet.

SIA

This displays the SIA Remote Control status.

GPS

GPS status (displays GPS if the GPS is enabled, otherwise displays nothing).

Logging: XXX

Logging status OFF or ON.

Events: XXX

Event reporting OFF or ON.

CAPS

Caps lock status.

NUM

Num lock status.

4.3 Multiple views

Splitter windows are used to display one or more views. Figure 4–6 illustrates the splitter view bar. Views are selected using the splitter view bar to split the main view screen into three primary views as depicted by the green shaded view icons. See Chapter 14 for more information on the pages loaded using the splitter view bar.

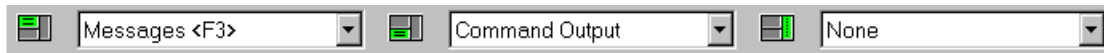


Figure 4–6 Splitter view bar

The splitter view bar has three pull-down menus. The left menu will always be displayed, and you can choose whether to display the other two.

Selections from the left menu, such as Messages <F3>, will display as a top view. Selections from the middle menu, such as Command Output, Memory Viewer, and Temporal Analyzer, will display as a bottom view. Selections from the right menu, such as NV Items, will display as a right-side view.

Examples of splitter view bar selections and the resulting splitter screens are shown in the following figures.

Figure 4–7 is an example of one splitter screen.

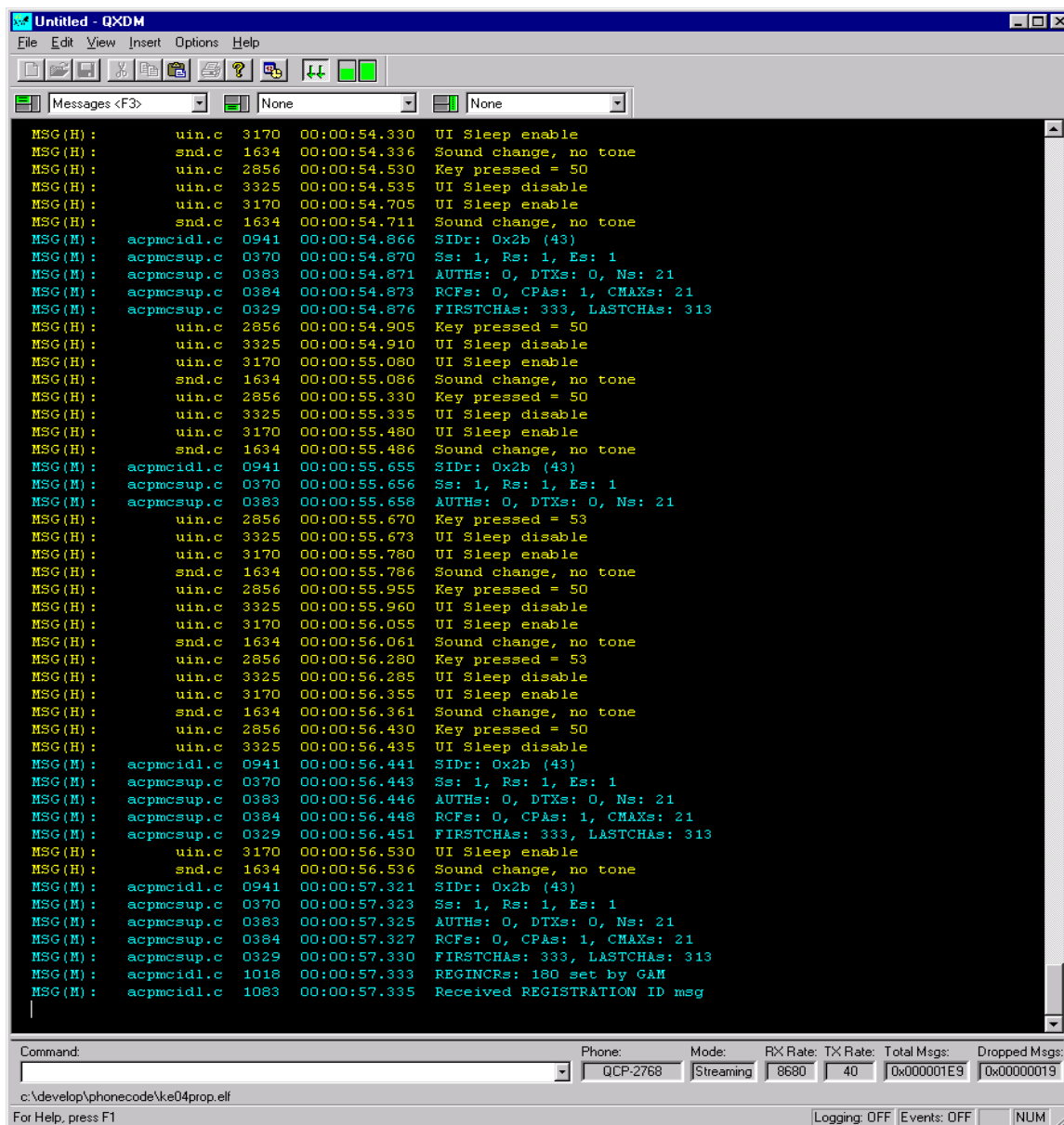


Figure 4–7 Example of one splitter screen

Figure 4–8 is an example of two splitter screens.

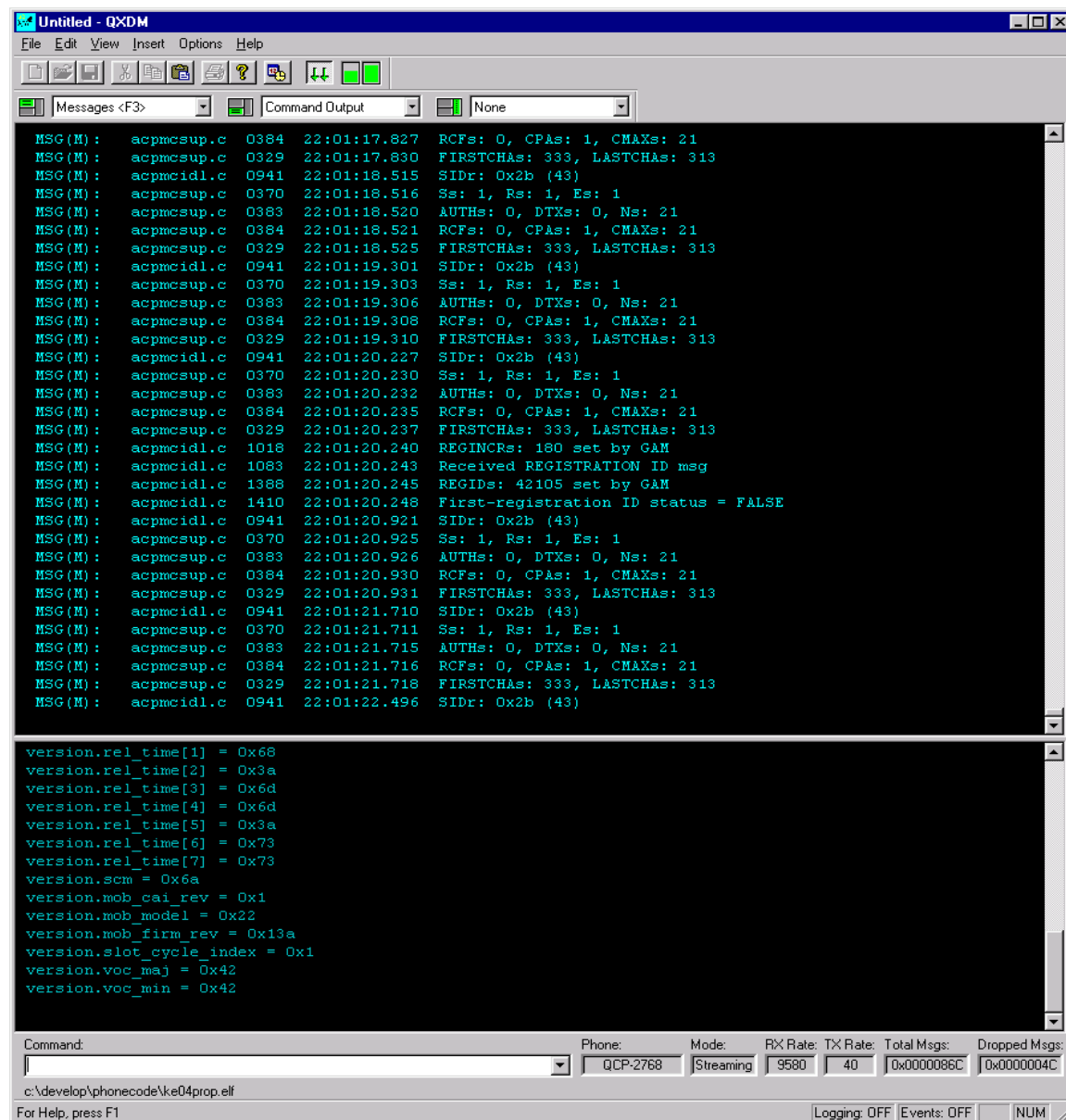


Figure 4–8 Example of two splitter screens

Figure 4–9 is an example of three splitter screens.

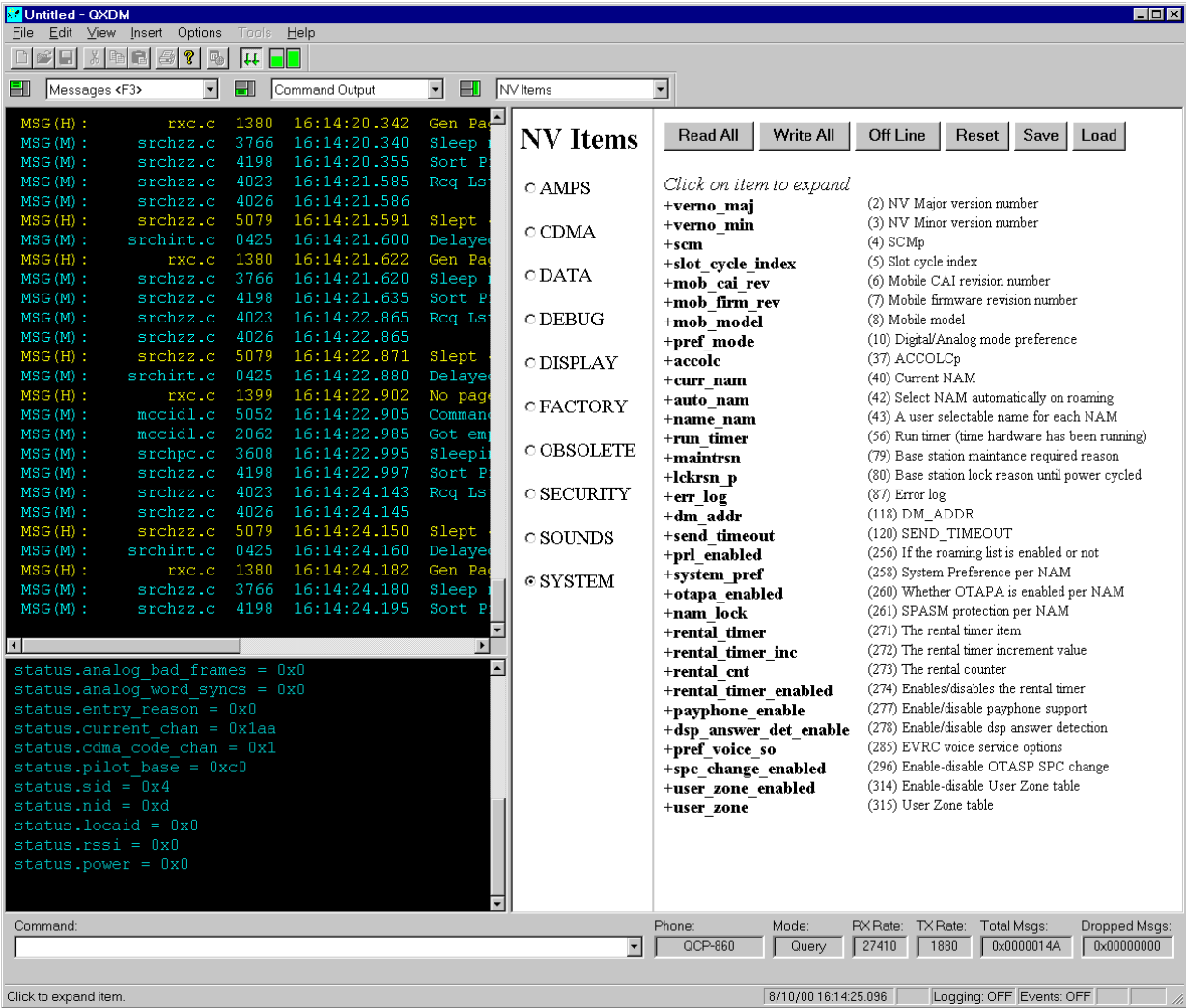


Figure 4–9 Example of three splitter screens

4.4 Additional views

The Command Output and Messages <F3> displays of QXDM show information from the SURF or FFA and use rich-edit text views, which provide text editing capabilities. Other properties are displayed using HTML browser views.

To view a page, select it from the drop-down menu associated with the desired splitter view and click the icon (see Figure 4–6).

4.4.1 Command Output view


To view output from commands entered on QXDM's command line, select Command Output from the middle menu.

4.4.2 Script Help view

You can select Script Help from any of the three menus.

4.4.3 Message view and Autoscroll button

Messages from the FFA or SURF are displayed in this view. To view Messages, select Messages <F3> from the left menu. Message rate, priority, and filtering are configured using the

Options → Settings → Messages dialog. In the toolbar there is an Autoscroll button  that will enable or disable auto scrolling without affecting addition of new messages.

4.4.4 Property view

The Property view displays of QXDM are written in HTML. You can select Property View from any of the three menus. QXDM acts as a browser to display HTML pages that reference data on the SURF or FFA. Pages can be modified or new pages added without requiring a rebuild of QXDM. See Chapter 6 for information on creating new properties for use with QXDM. Chapter 14 describes the Property views currently shipped with QXDM.

4.4.5 Memory viewer

To view the memory viewer, as shown in Figure 4–10, select Memory Viewer from the middle menu. The memory viewer provides peek and poke into the subscriber memory. This display allows you to view and edit memory locations inside the phone at run-time. To edit memory, modify the contents and then press ENTER or ESC.

00040000	0C DC 03 28 0A DC 00 22 00 21 00 20 07 B4 13 4B	.Ü...Ü.....K
00040010	39 1C 13 4A 03 20 0F F0 17 FE 03 B0 39 1C 10 48	9..J...ð.p..9..H
00040020	E6 F7 E8 FE 13 E0 01 24 F8 1D 19 30 04 70 19 21	æ.èþ.â..ø..0.p..
00040030	01 72 FF 21 41 72 13 23 5B 01 F8 18 04 70 38 1C	.ry.Ar....ø..p8.
00040040	FF F7 DC F9 38 1C 28 F0 73 FD 06 48 04 70 90 BD	y.Üù8..ðsy.H.p..
00040050	44 33 02 01 BC 0C 00 01 3C AE 0C 00 B4 B1 0C 00	D3.....
00040060	1B 05 00 00 F9 0A 00 01 90 B5 07 1C 24 48 04 1Cù.....H..

Figure 4–10 Memory viewer

5 Logging

Logging is enabled/disabled by pressing ALT+L or by selecting Options → Logging. To edit the log mask, use the Options → Log Mask dialog; you may also display this dialog by pressing F5. Figure 5–1 depicts the dialog.

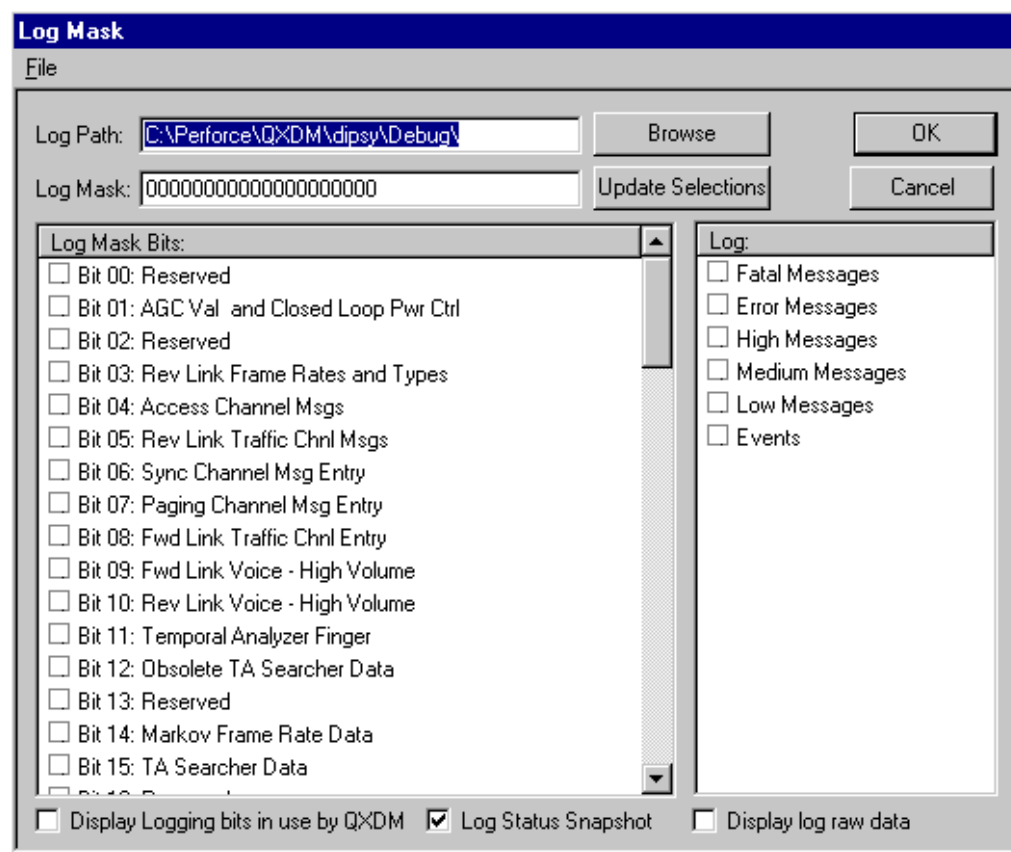


Figure 5–1 Log Mask Configuration dialog

1 If you change the log mask by typing a log mask string in the Log Mask dialog, you must click
2 **Update Selections** after entering the new log mask.

3 The number of bits displayed in the Log Mask dialog will vary depending on the number of bits
4 supported by the connected FFA or SURF. To configure the names of the log mask bits, edit the
5 logmask.txt file located in the property\qxdm directory in the QXDM installation directory.

6 Select the Display Logging bits in use by QXDM check box to highlight the log bits that are in use
7 due to display views you may have showing.

8 Select the Log Status Snapshot check box to enable logging of the tool-generated Status snapshot
9 packet.

10 Select the Display log raw data check box to display the contents of log packets to the
11 Messages <F3> screen.

12 If a screen that uses logging data is loaded, such as Status or Temporal Analyzer, the logging bits
13 required by these screens cannot be changed by you. These screens must first be unloaded in order to
14 change these bits. You may still disable logging entirely while these screens are loaded.

15 You can also specify the location on your machine where you want the log files saved. If the log path
16 is empty, the log files will be saved in the current directory by default.

17

6 Properties

QXDM uses properties to access data in the phone. Properties provide a way for the user to describe target data structures to QXDM dynamically.

Some properties are defined using a property file. Other properties are defined by the DWARF server (see Chapter 10 for more information) and do not have a corresponding property file. In either case, access to properties is typically performed using HTML scripts. The other approach is to use the `list_properties`, `get_property`, `print_property`, and `put_property` commands on QXDM's command line, as shown in Figure 6–1. Use `get_property` to read data and `put_property` to write data to the phone. The `print_property` command displays the property definition that QXDM is using. This is helpful as a lookup for property field names and when debugging user-generated properties.

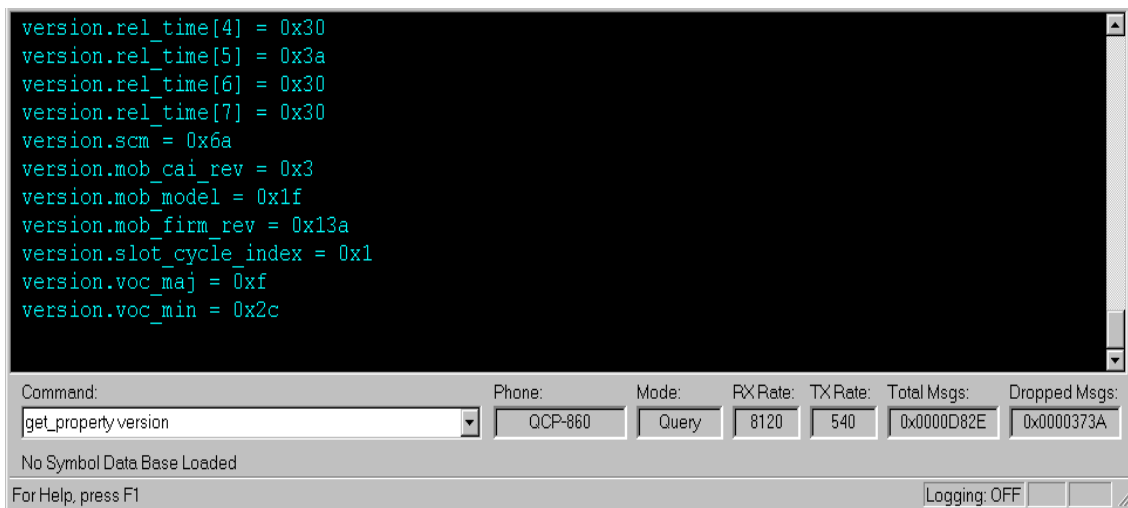


Figure 6–1 Command line access to properties

Listing properties

The list of current properties or properties matching a given pattern can be displayed using the following syntax:

```
list_properties [part of a property name]
```

[part of a property name] is optional and will result in a list of all current properties matching the part specified.

Reading property data

Reading property data is done using the following syntax:

```
get_property <PropertyName>
```

<PropertyName> is the name of the property to be read. Typing `get_property` before the property name is optional for all base property names except NV Items.

Writing property data

Writing property data is done using the following syntax:

```
put_property <PropertyName> <PropertyArguments>
```

<PropertyName> is the name of the property to be written. Typing `put_property` before the property name is optional for all base property names except NV Items. <PropertyArguments> refers to all property field values separated by spaces or commas with spaces.

Printing property definitions

Printing a property definition is done using the following syntax:

```
print_property <PropertyName>
```

<PropertyName> is the name of the property to be listed.

6.1 Property definition

Properties are defined as shown in Table 6-1. Comments may be added using the # character for properties defined in property files. Anything following a # is assumed to be a comment and is ignored. Each noncomment line of a property file will contain the following information (square brackets indicate optional parameters):

```
Property.Element, Offset, Size [, Description, Data Type, Command/Item
enum/Address]
```

Table 6-1 Property file information

Item	Description
Property	Name of the property
Element	Name of an element within the property
Offset	Offset of the element from the start of the packet
Size	Size in bytes of the element
Description	Description of the element
Data type	One of the following data types: INT, UINT, CHAR, HEX
Command/item enum/address	Command or Item ID for a packet definition or data address

6.2 Property packets

The following properties are defined as packets. Refer to the *CDMA DMSS Serial Data Interface Control Document*, 80-V1294-1 X6, specific Application Note documents, or Compatibility Notes for the actual packet descriptions.

Diagnostic property packet

The element name `cmd_code` is always the first element of any diagnostic property packet. The line containing the element `cmd_code` must also have the actual Command ID of the diagnostic packet as the sixth parameter, as shown in the following example:

```
Markov.cmd_code, 0, 1, , , 26
```

The offset of the `cmd_code` is always 0, and the size is always 1. The sixth parameter is the Command ID (shown as 26 in the Markov example).

Log property packet

The element names `length`, `log_code`, and `time_stamp` are always the first three elements of any log property packet. The line containing the element `log_code` must have the actual log code of the log packet as the sixth parameter:

```
genta.length,          0,  2
genta.log_code,        2,  2,,, 25
genta.time_stamp,      4,  8
```

The offset of the `log_code` is always 2, and the size is always 2. The sixth parameter is the Log Code (shown as 25 in the General Temporal Analyzer genta log packet example).

NV Item property packet

The element name `nv_item` is always the first element and `nv_stat` is always the last element of any NV Item property packet. The line containing the element `nv_item` must have the actual NV Item number of the NV Item packet as the sixth parameter:

```
esn.nv_item,          0,  2,,, 0
esn.esn,              2,  4
esn.nv_stat,          6,  2
```

The offset of the `nv_item` is always 0, and the size is always 2. The sixth parameter is the NV Item number (shown as 0 in the `nv_esn` NV item example). The offset of `nv_stat` is based upon the total size of the NV Item. In the `nv_esn` example, the offset is 6. The size of `nv_stat` is always 2.

6.3 Absolute address on the phone

The `abs_addr` element is always the first element of any absolute address property definition. The line containing the element `abs_addr` must have the actual address of the data as the sixth parameter. If the address is written in hexadecimal, then `0x` must be prepended to the address as shown below:

```
MSM5000_DEMOD_RESET.abs_addr,    0,  4,,, 0x3000000
MSM5000_DEMOD_RESET.length,      4,  2
MSM5000_DEMOD_RESET.data,        6,  4
```

6.4 Properties with arrays

The recommended representation of array elements is to specify each array element. For example, an array of two elements, each 4 bytes in size (for example, dword array[2]), is defined as (assume 20 bytes of additional elements precede `packet_name.array`):

```
packet_name.array,           20,  8
packet_name.array[0],        20,  4
packet_name.array[1],        24,  4
```

If the array is defined this way, you may then access the array elements, from JavaScript, in two ways:

```
GetProperty("packet_name.array[0]");
GetProperty("packet_name.array[1]");
```

or

```
GetPropertyEx("packet_name.array", 0, 4);
GetPropertyEx("packet_name.array", 4, 4);
```

Using the `GetPropertyEx()` method allows easier access to array elements. For example:

```
arraySize = GetSize("packet_name.array");
elementSize = GetSize("packet_name.array[0]");
for (ii = 0; ii < arraySize; ii += elementSize)
{
    data = GetPropertyEx("packet_name.array", ii, elementSize)
    ....
}
```

6.5 Structures in properties

All text before the first dot (‘.’) in the name is considered the property name. All text after the first dot in the name is considered the element name. Therefore, nested structures can be represented by a list of elements separated by dots. Consider the following example of a structure embedded within a packet:

```
struct structure
{
    word wordItem,
    byte byteItem,
    long longItem
}
```

(assume 20 bytes of additional elements precede packet_name.structure)

```
.....
packet_name.structure.wordItem,      20,   2
packet_name.structure.byteItem,      22,   1
packet_name.structure.longItem,      23,   4
```

With data structures, defining the following would not be useful and is not recommended:

```
packet_name.structure, 20, 7
```


6.6 Property support tools

Two tools are provided with QXDM to simplify the task of building property files and HTML scripts. Perl from ActiveState, version 5.005_03 or newer, and Microsoft C++ 6.0 or newer are required to run the tools. They are located in the qxdm\property\user directory where QXDM is installed.

It is important to ensure that both the Microsoft VC++ 6.0 compiler and Perl run from the MS-DOS command shell. It should be possible to run these tools and obtain results similar (may differ slightly on your machine) to the following:

```
C:\> cl
Microsoft (R) 32-bit C/C++ Optimizing Compiler Version 12.00.8168 for 80x86
Copyright (C) Microsoft Corp 1984-1998. All rights reserved.
usage: cl [ option... ] filename... [ /link linkoption... ]

C:\> perl -v
This is perl, v5.6.1 built for MSWin32-x86-multi-thread
(with 1 registered patch, see perl -V for more detail)

C:\Program Files\Qualcomm\QXDM\property\user>perl makeProp.pl -help
MakeProp
Make Property File
Creates .prop file from C header file.
Options:  -I=<path>           Search include paths
...
```

If the results are not similar, makeProp and makeHTML will fail.

6.6.1 makeProp tool

makeProp creates a property file (.prop) from a C header file containing typedef structs.

Syntax: makeProp [options] header

Options:

- I=<path> Search include paths
- nvitems Build NV item property file from the DMSS nv.h.
- cmd=<Command Code> Use <Command Code> value
- item=<Item Name> Use <Item Name>
- types=<type(s)> Get list of types from a C header file. Enclose types in double quotes if more than one type or use a response file (preceded with @).
- diagPkt Build diag packet property file.

Examples: makeProp.pl -nv nvTypes.h -I=C:\dmss
 makeProp.pl -cmd=26 -item=markov markovStruct.h
 makeProp.pl diagpkt.h -diagpkt -types=diag_password_req_type

makeProp.pl is included with QXDM to aid in generating properties from C typedef structs, but is very limited due to the text processing nature of the tool.

Using a new property

To enable QXDM to use a new property file, copy it to the QXDM\Property\qxdm folder where QXDM was installed and restart QXDM.

6.6.2 makeHTML tool

makeHTML creates an HTML file from a property file that was created by makeProp.

Syntax: makeHTML [options] property_file ...

Options:

- h Display this message.
- f=<html file> Write all HTML output to <html file>. Default is to write a separate HTML file for each property contained in the input file.

Example: makeHTML.pl my_data.prop -f=my_data.html

Using a new HTML display

To enable QXDM to use a new HTML display, copy it to the QXDM\HTML folder where QXDM was installed and restart QXDM.

7 Events

1 Events are properties that have additional features described in this chapter.

2 Events are enabled or disabled by pressing **ALT+E** or by selecting **Options→Event Reporting**. When
3 enabled, events are displayed in the Messages <F3> window or in other displays containing events.
4 Note that event reporting is automatically enabled when you bring up any event displays even when
5 the Event Reporting option is turned off. Event reporting is automatically disabled when there are no
6 event displays and the Event Reporting option is turned off.

7 7.1 Event information

8 All event information such as event name, event payload parsing format, event/category relationship,
9 and so on, is stored in the QXDM Property database and can be modified using the QXDM Property
10 Editor. See Chapter 8 for further details.

11 7.2 Event display

12 There are two types of event-based display: scrolling display and table display.
13

7.2.1 Scrolling display

The scrolling display is similar to the Messages <F3> screen in which the events are scrolled down and sorted by the timestamp. You can right-click the display to bring up the Event Options menu. From this options menu the display can be configured and the autoscrolling feature can be paused. See Section 7.3 for further details on configuring the display.

You can view up to three event displays simultaneously by using all three splitter windows.

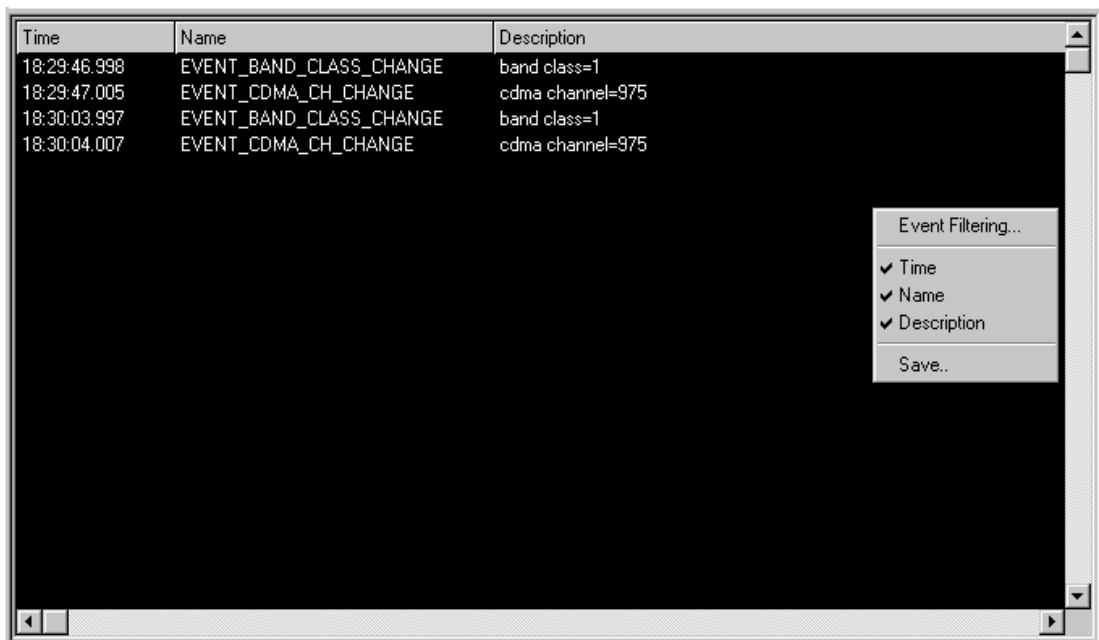
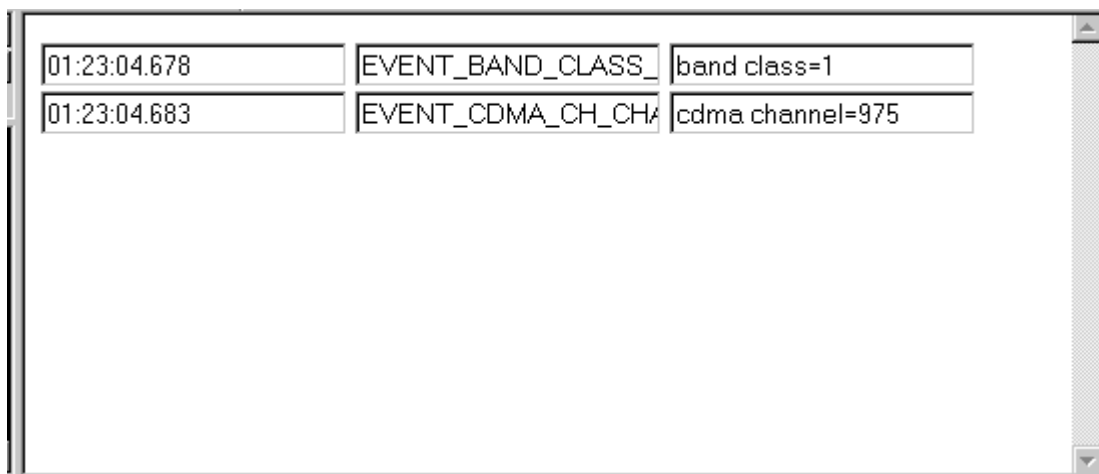


Figure 7–1 Scrolling event display

7.2.2 Table display

This type of display shows event information in a tabular format. The event data is refreshed when new events arrive.



01:23:04.678	EVENT_BAND_CLASS	band class=1
01:23:04.683	EVENT_CDMA_CH_CHA	cdma channel=975

Figure 7–2 Table event display

The display is HTML-based. The example below shows how to write a table event display such as the one in Figure 7–2.

```
<!NAME="HDR State Status" WIDTH=450 HEIGHT=290>
<HTML>
  <HEAD>
    <TITLE> HDR State Status</TITLE>
    <STYLE TYPE="text/css">
      DIV {font-size:11pt}
      SPAN {font-size:11pt}
      .expandable {color:black; cursor:hand}
      .expanded {color:blue; margin-left:10pt}
      .collapsed {display:none}
    </STYLE>
  </HEAD>
  <SCRIPT LANGUAGE="JavaScript">
    HTML_PAGE_NAME = " HDR State Status";
```

```
1
2      // Check whether IE4 or later.
3      var MS = navigator.appVersion.indexOf("MSIE");
4      window.isIE4 = (MS > 0) &&
5          (parseInt(navigator.appVersion.substring(MS + 5, MS + 6)) >= 4);
6  </SCRIPT>
7  <SCRIPT LANGUAGE="JavaScript" SRC="IDispatch.js"></SCRIPT>
8
9  <BODY>
10     <INPUT TYPE="text" ID=EVENT_BAND_CLASS_CHANGE NAME="Timestamp" VALUE="timestamp">
11     <INPUT TYPE="text" ID=EVENT_BAND_CLASS_CHANGE NAME="Name" VALUE="name">
12     <INPUT TYPE="text" ID=EVENT_BAND_CLASS_CHANGE NAME="Description" VALUE="description">
13     <BR>
14     <INPUT TYPE="text" ID=EVENT_CDMA_CH_CHANGE NAME="Timestamp" VALUE="timestamp">
15     <INPUT TYPE="text" ID=EVENT_CDMA_CH_CHANGE NAME="Name" VALUE="name">
16     <INPUT TYPE="text" ID=EVENT_CDMA_CH_CHANGE NAME="Description" VALUE="description">
17     <BR>
18  </BODY>
19 </HTML>
20
```

21 The code in the HTML body describes how we specify which events are to be filtered and which
22 columns are to be displayed.

7.3 Event options

This feature allows users to configure the scrolling event display.

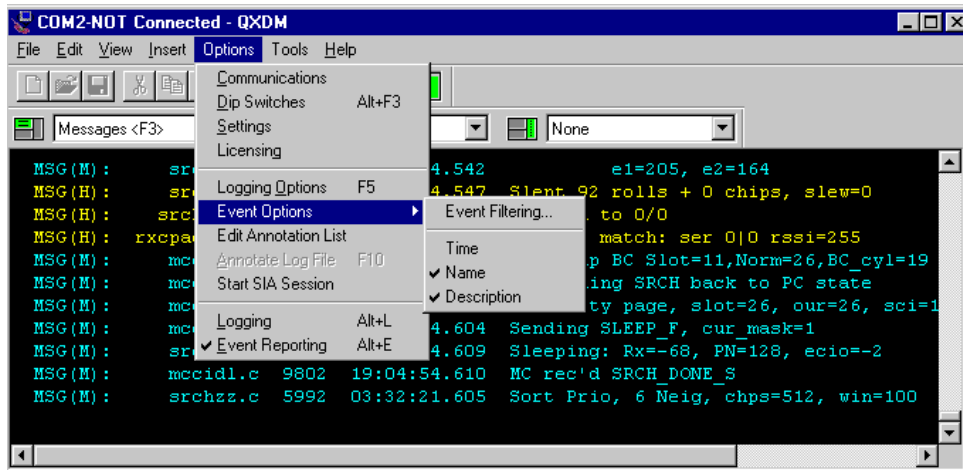


Figure 7–3 Options → Event Options

Select Options → Event Options from the QXDM menu.

- Event Filtering – filters the events and/or event categories you want to see in the displays. Select this to bring up the Event Filtering Dialog. See Section 6.3 for further details
- Time – displays/hides the Timestamp column
- Name – displays/hides the Name column
- Description – displays/hides the Description column

The Event Options setting that you select from the main menu has a global effect. This means that the configuration is applied to all the event displays that are subsequently brought up. However, it will not affect the ones currently displayed.

To change the configuration of a particular event display, right-click the display to bring up the options menu. The configuration change is only applied locally to this display and does not affect other event displays.

7.4 Event filtering

Event Filtering allows QXDM users to select which events and event categories they want to see in the display. This is illustrated in Figure 7–4.

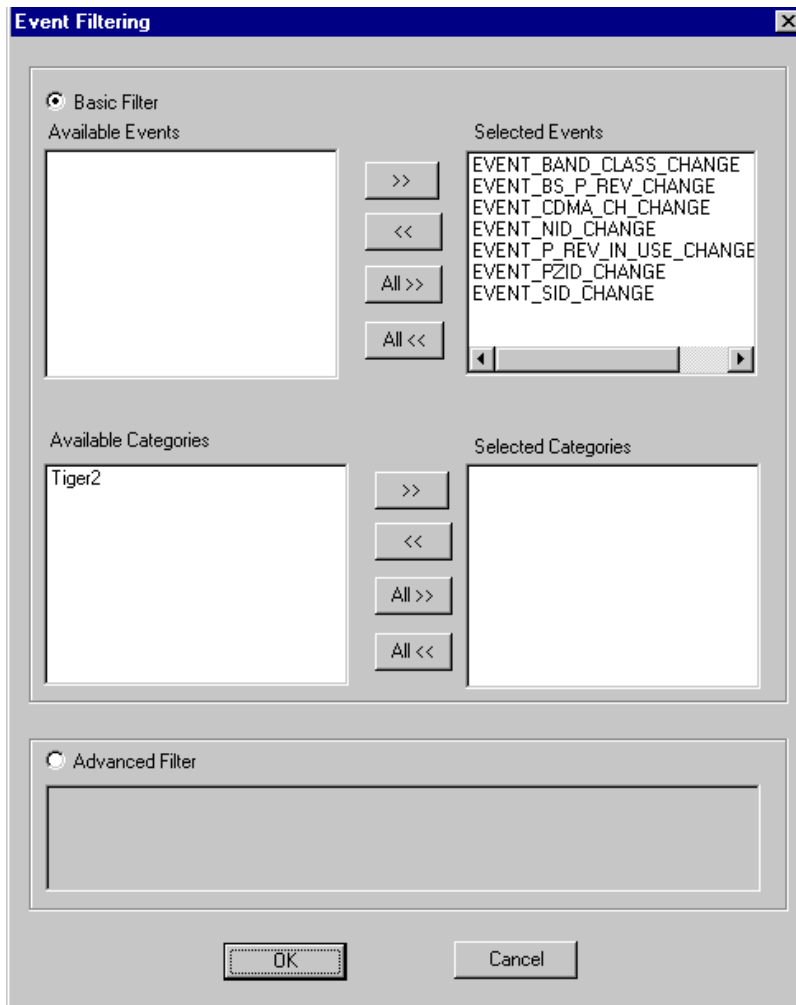


Figure 7–4 Event filtering

7.4.1 Basic event filtering

To use the basic filtering, select the Basic Filter option in the Event Filtering dialog screen.

Use the buttons to select the events and categories you want to see in the event displays. Category is used to conveniently group similar events together, so that you can filter a group of events instead of each event individually. The event/category relationship is defined in the QXDM database, and can be modified using the QXDM Property Editor. See Chapter 8 for further details.

7.4.2 Advanced event filtering

This feature allows users to perform more complex filtering by issuing an SQL query directly to the database.

To use the advanced filtering, select the Advanced Filter option in the Event Filtering dialog screen, and then enter the SQL-like query. The query should follow the SQL syntax as in the WHERE clause. Consult appropriate documents for SQL syntax.

Note that:

- Can refer to an event by ID or by name
- Have to use double quotes for a string, such as an event name or category

Here are some examples using Tiger 2 events to illustrate how to use this feature:

1. To select event EVENT_BAND_CLASS_CHANGE, enter ID=256

Or you can enter:

Name="EVENT_BAND_CLASS_CHANGE"

2. To select event EVENT_BAND_CLASS_CHANGE and event EVENT_CDMA_CH_CHANGE, enter ID=256 or ID=257

Or you can enter:

ID=256 OR Name="EVENT_CDMA_CH_CHANGE"

3. To select all events in Tiger2 category, enter CategoryName="Tiger2"

4. To select all events in Tiger2 category, except event EVENT_BAND_CLASS_CHANGE, enter:

CategoryName="Tiger2" AND NOT Name="EVENT_BAND_CLASS_CHANGE"

Or you can enter:

CategoryName="Tiger2" AND NOT ID=256

5. To select all events in Tiger2 category, except those in Category1, enter CategoryName="Tiger2" AND NOT CategoryName="Category1"

7.5 Custom event display

This feature provides QXDM users the flexibility to create their own customized event displays and save them in text-based XML format. XML is a markup language concerned with the description and structuring of data, and is fast becoming the language of choice for data storage.

The following is an example of an event display in XML format is:

```
<Display>
  <Name>HDR State</Name>
  <Column>Timestamp</Column>
  <Column>Name</Column>
  <Event>EVENT_BAND_CLASS_CHANGE</Event>
  <Event>EVENT_BS_P_REV_CHANGE</Event>
  <Event>EVENT_CDMA_CH_CHANGE</Event>
  <Event>EVENT_NID_CHANGE</Event>
  <Event>EVENT_P_REV_IN_USE_CHANGE</Event>
  <Event>EVENT_PZID_CHANGE</Event>
  <Event>EVENT_SID_CHANGE</Event>
  <Category>Tiger2</Event>
  <BasicFilter>0</BasicFilter>
  <Query>ID=101 OR Name="EVENT_1"</Query>
</Display>
```

There are two ways to create a custom event display:

1. Use the QXDM GUI. Bring up a generic Events display from the QXDM menu. Right-click the display to configure the filtering and select the columns to be displayed. Select Save and enter a name to save the selected configuration into a new custom event display. Note that the name has to be unique or you would get an error. The new display is dynamically loaded into QXDM, so you can select it from the QXDM menu.
2. Modify the XML file directly. Make sure the change is syntactically correct by running the XML file through an XML parser. Microsoft Internet Explorer 5 has a built-in XML parser, so just make sure you can open the file successfully with IE5. QXDM also logs parsing errors if there are any when it loads the XML file. Note that the changes are not dynamically loaded to QXDM, so you have to restart QXDM.

There are two XML files:

- - \property\QXDM\Displays.xml – This is the official XML file to be shipped with every QXDM release, and is overwritten during the installation.
- - \property\user\UserDisplays.xml – This is the user XML file; it will not be overwritten during the installation. The custom displays you create via the QXDM GUI are stored in this file. If you want your custom displays to be official and shipped with the QXDM release, you must submit your displays to the Tools team so that they can be included in the official XML file.

8 Property Editor

1 QXDM stores Event and Property definitions in a database (*.qdb) format. QXDM uses two of these
2 database files, the QXDM built-in database and the user database. The QXDM Property Editor allows
3 you to edit the contents of the user database. The Property Editor can be accessed from Start →
4 Programs → QXDM → QXDM Property Editor.
5

NOTE The Property Editor treats Events as a type of Property. So in this chapter, the term “Property”
includes Events.
6

8.1 Built-in Property database

8 The QXDM Built-in Property database is found in the Property\QXDM folder in your QXDM
9 installation. This database contains officially published Properties that are distributed to all QXDM
10 users. The contents of this database changes between releases of QXDM. QUALCOMM recommends
11 that you do not modify the contents of this database. The Property Editor allows you to browse the
12 contents of this database but not modify its contents.

13 The Properties and Categories from the Built-in database are displayed next to dark, grayed icons to
14 indicate that these are read-only entities.

8.2 User Property database

16 The User Property database is found in the Property\User folder in your QXDM installation. This
17 database is empty in a new installation. This database will contain the Properties that are created by
18 you locally. The Property Editor automatically stores user-defined Properties in this database. It
19 allows you to modify the contents of this database.

20 The Properties and Categories from the User database are displayed next to light, nongrayed icons to
21 indicate that these can be modified or deleted.

8.3 Property Editor User Interface

The Property Editor application has an Explorer-style splitter window user interface, as shown in Figure 8–1.

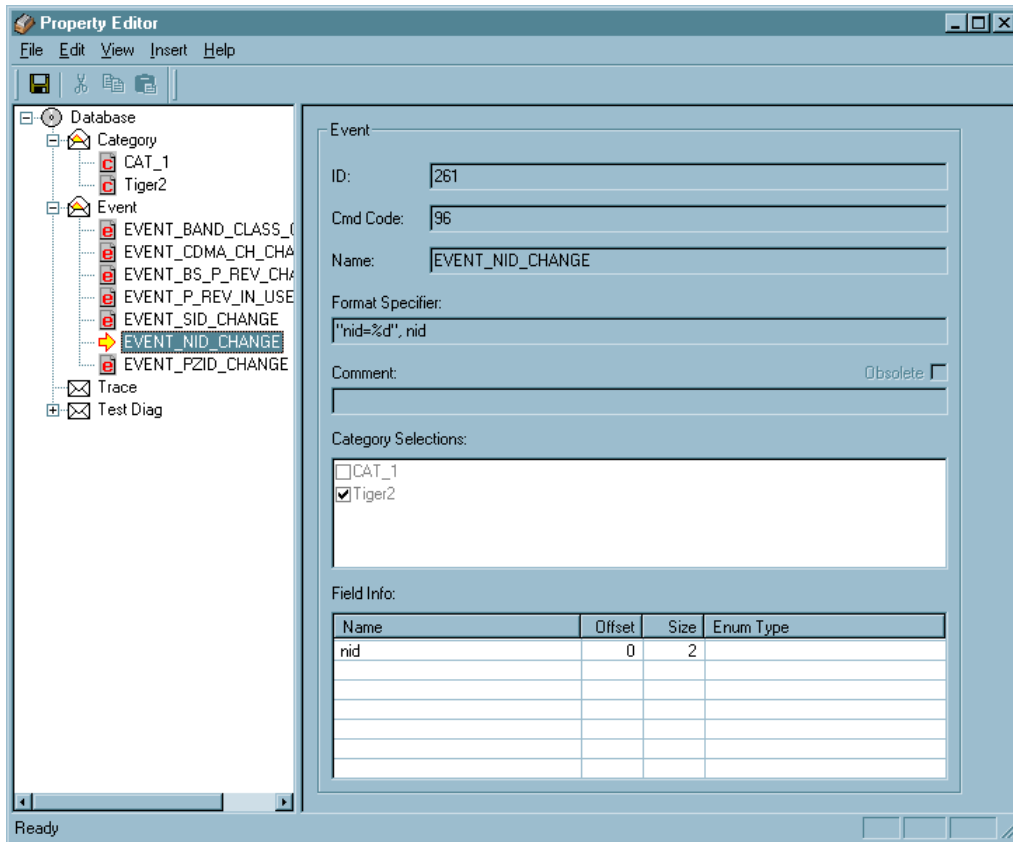


Figure 8–1 Property Editor window

8.3.1 Splitter Window Overview of the Property Editor

The left panel of the application window displays the Properties and Categories in a tree format. Each Property type is displayed as a folder in the tree. The tree also has a special folder containing all the Categories in the database.

The individual properties are listed under the appropriately named folder. Selecting an individual Property or Category in the left panel displays more information about the selected item in the right panel of the splitter window. The right panel of the splitter window is empty if a folder or the database root node is selected in the left panel.

8.3.2 Category Display of the Property Editor

Figure 8–2 depicts the Category display splitter window.

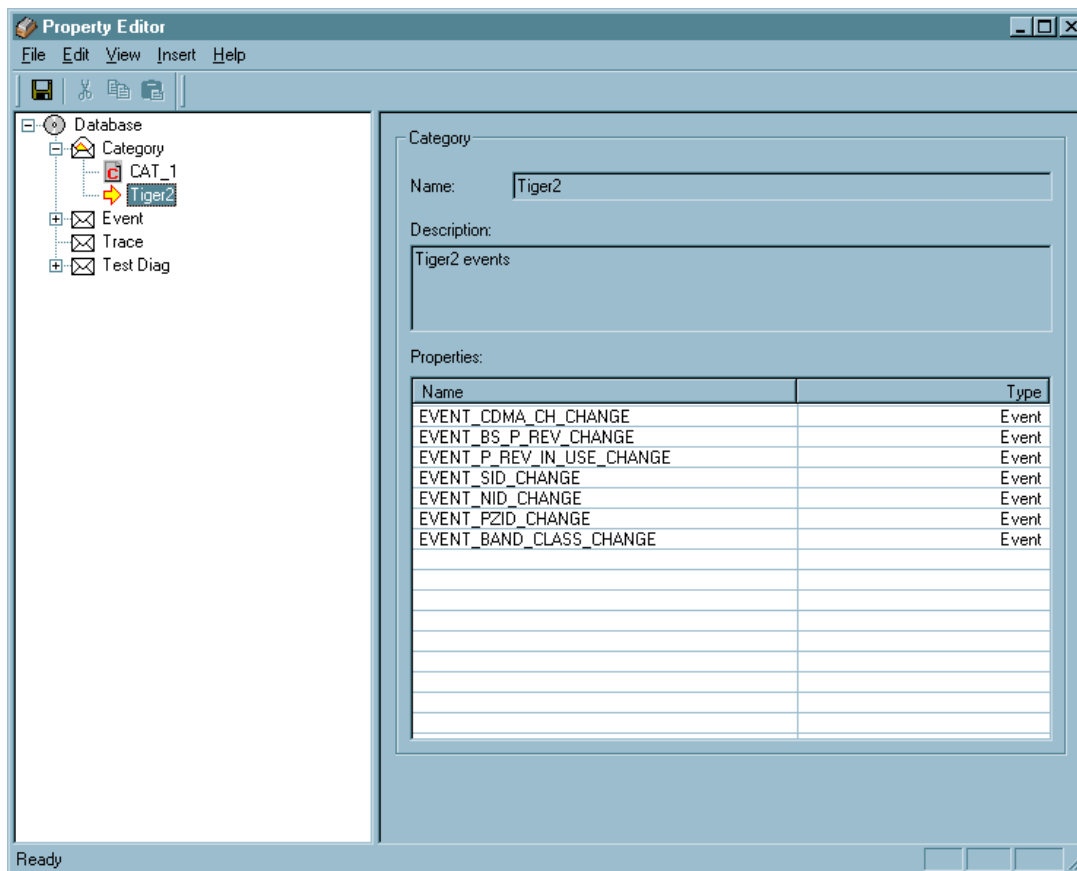


Figure 8–2 Category display

The Category display on the right panel shows the name of the Category, its description, and a table containing all the Properties that belong to the specified Category, along with their types. If Category is user-defined, the Description field can be edited. The Properties table is read-only. To change the Category assignment for an individual Property, use the display for that Property.

8.3.3 Property Display of the Property Editor

Figure 8–3 depicts the Property display splitter window.

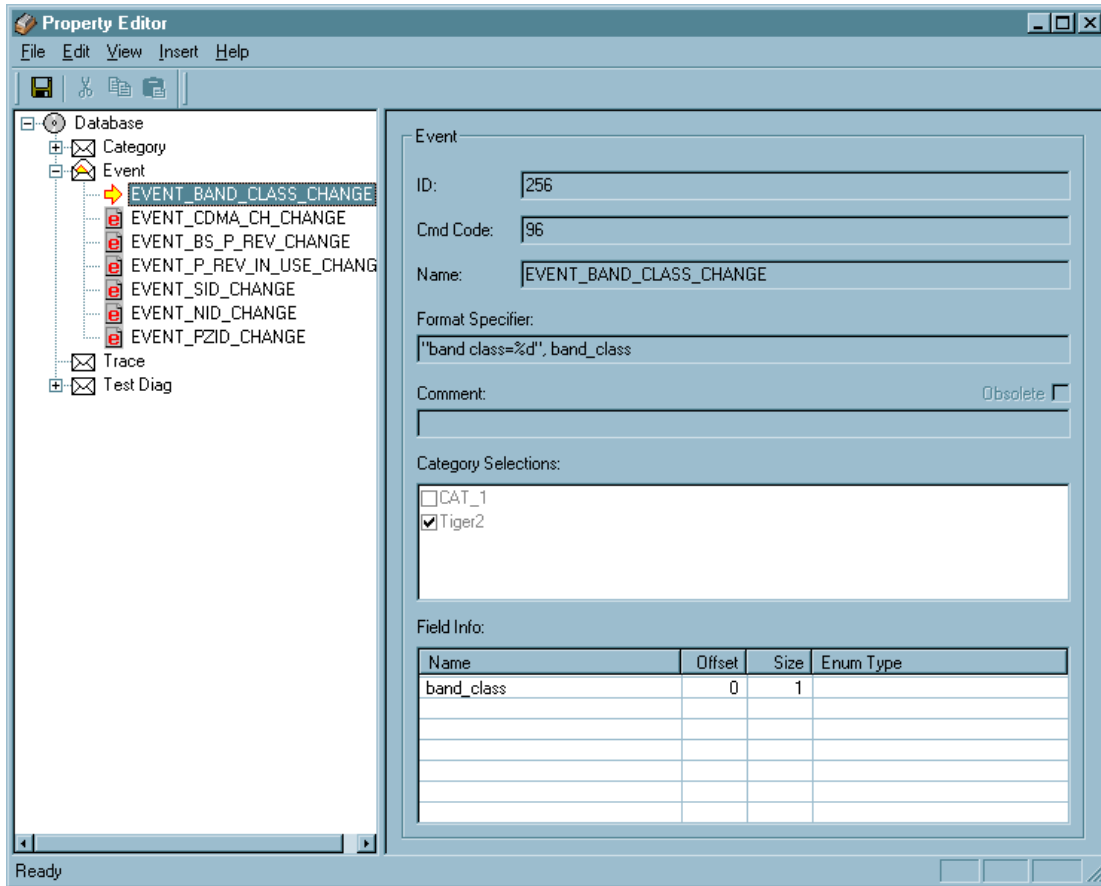


Figure 8–3 Property display

The Property display on the right panel shows the type of the Property, its ID, command code, name, format specifier string for QXDM displays, and a table containing the Property Field organization for the given Property. The display also contains a check list for the Categories with the relevant Categories selected. A check box titled Obsolete indicates if the given Property is obsolete. There is also a Comment edit field.

For user-defined Properties, you are allowed to modify the Category Selections, Format Specifier, and the Comment fields. The Property Editor also provides a facility to edit the Field assignments for the Property.

8.4 Working with properties

The Property Editor does not allow you to modify or delete built-in Properties. The following sections describe the steps to work with user-defined Properties.

8.4.1 Inserting a new property

To insert a new property:

1. Select Insert→Property or press F3.

This brings up the select Property dialog, as shown in Figure 8–4.

2. Select the type of the new Property to be inserted and click **OK**.

A new Property is created with a default name under the folder that corresponds to the selected Property type.

3. Rename the new Property as desired.



Figure 8–4 Select Property dialog

8.4.2 Renaming a property

To rename a property:

1. Select the Property to be renamed.
2. Select Edit→Rename or press CTRL+R.

This makes the selected left-panel node editable (see Figure 8–3 left panel).

3. Type in the new name for the Property.
4. Press ENTER.

8.4.3 Deleting a property

To delete a property:

1. Select the Property to be deleted.
2. Select Edit→Delete or press DELETE.

This removes the selected left-panel node (see Figure 8–3 left panel).

8.4.4 Modifying category selections

In the Category Selections check list (shown in Figure 8–3 right panel), check the desired Categories and uncheck the others.

8.4.5 Modifying property field structure

Double-click the Field Info table (shown in Figure 8–3 right panel) to invoke the Field Edit dialog. See Section 8.6 for information on working with Property fields.

8.5 Working with categories

The Property Editor does not allow you to modify or delete built-in Categories. The following sections describe the steps to work with user-defined Categories.

8.5.1 Adding a new category

To add a new category:

1. Select Insert→Category or press F2.

A new Category is created with a default name under the Category folder.

2. Rename the new Category as desired.

8.5.2 Renaming a category

To rename a category:

1. Select the Category to be renamed.
2. Select Edit→Rename or press CTRL+R.

This makes the selected left-panel node editable (see Figure 8–2 left panel).

3. Type in the new name for the Category.
4. Press ENTER.

8.5.3 Deleting a category

To delete a category:

1. Select the Category to be deleted.
2. Select Edit→Delete or press DELETE.

This removes the selected left-panel node (see Figure 8–2 left panel).

8.6 Working with property fields

Double-clicking the Property Field table of a user-defined Property invokes the Field Edit dialog, as shown in Figure 8–5.

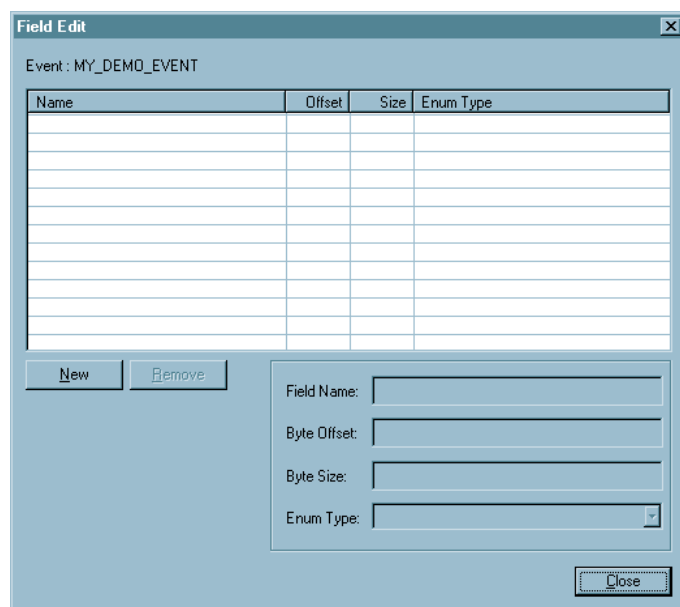


Figure 8–5 Field Edit dialog

The Field Edit dialog displays the type and name of the Property being edited. It also contains a table similar to the Field Info table in the right panel of the Property display (shown in Figure 8–3).

8.6.1 Adding a new property field

To add a new property field:

1. Click **New** on the Field Edit dialog.

This inserts a new Property Field with a default name and field values.

2. Change the individual values as described in Sections 8.6.2 and 8.6.3 below.

8.6.2 Modifying property field contents

To modify property field contents:

1. Select the Property Field to be modified.

The contents of the selected Property Field are displayed in the Edit Fields at the bottom of the dialog.

2. Modify the values as desired.

This propagates the change to the display table.

8.6.3 Deleting a property field

To delete a property field:

1. Select the Property Field to be deleted.
2. Click **Remove** on the Field Edit dialog.

8.7 Saving the database

Modifications from the Property Editor are not committed to the database files automatically. The changes have to be saved to the database when required. To save the contents, select File→Save. The changes made are saved in the User Property database. The QXDM built-in database is left unchanged.

If you try to exit the Property Editor application without saving the changes, the application reminds you about the changes and confirms if you want to discard them.

8.8 Conflicts

Due to the periodic updates to the QXDM built-in database, conflicts may appear in the Property database. A conflict occurs when a new built-in database Property has the same ID or name as an existing user database Property. This happens because Property IDs are sequentially assigned by the Property Editor and the official Property publication mechanism.

It is important to check for conflicts each time a new version of QXDM is installed. Conflicts occur only if there are user-defined contents in the database. Use the following ways to verify if there are conflicts in the database:

- In QXDM, most Event or Property displays warn you of conflicts in the database.
- Open Property Editor. The opening screen displays a conflict warning message, as shown in Figure 8–6.

CAUTION All existing conflicts must be resolved before making any changes to the databases. Failure to do so will lead to an inconsistent state in the databases. This affects the QXDM displays that depend on the Property database. These displays may be erroneous or may even cause QXDM to hang or crash.

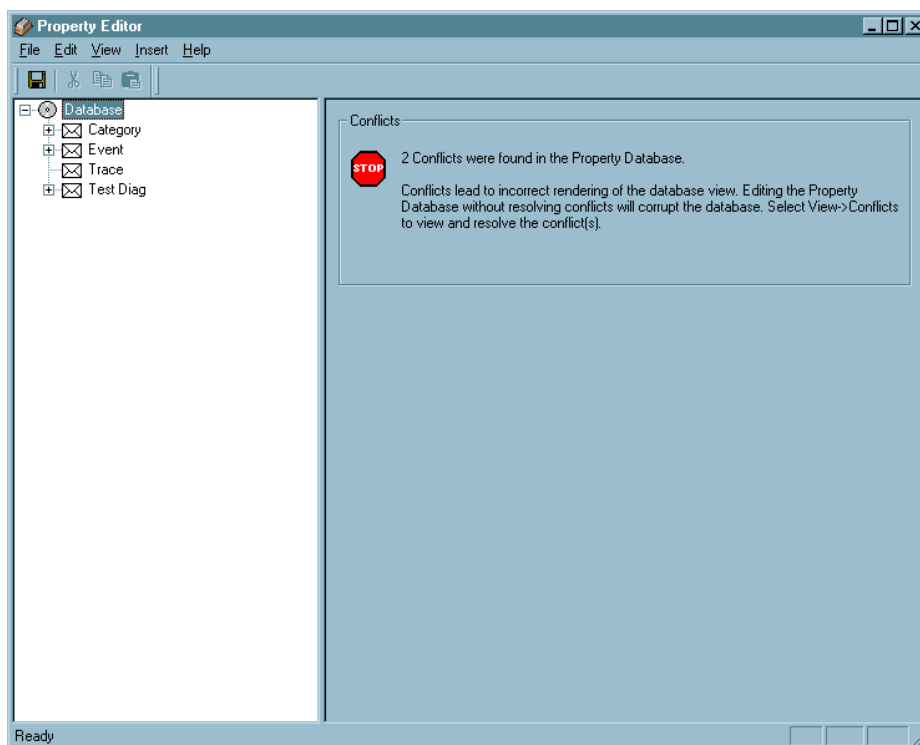


Figure 8–6 Property Editor conflict warning

8.8.1 Resolving conflicts

The Property Editor provides an extensive auto resolution feature for conflicts. To resolve conflicts, select View→Conflicts or press F5. This invokes the Conflict List dialog, as shown in Figure 8–7.

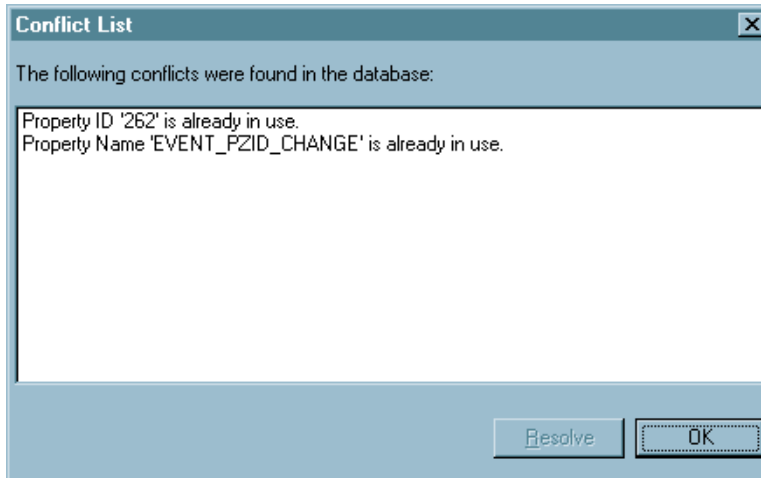


Figure 8–7 Conflict List dialog

The conflicts in the Conflict List are selectable. Select each conflict in this list and click **Resolve**. This brings up the Conflict Resolution dialog.

If the conflict is between the IDs of two Properties, the Resolution dialog allows you to auto generate a nonconflicting ID or delete the conflicting user-defined Property, as shown in Figure 8–8.

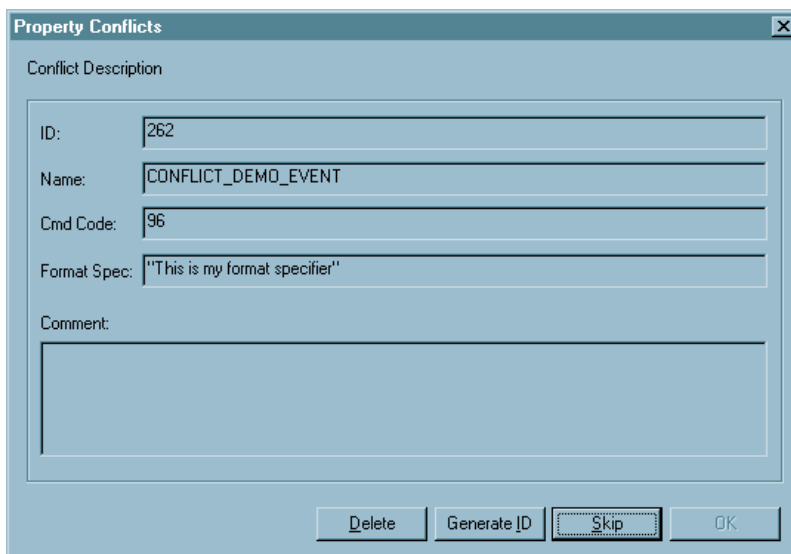
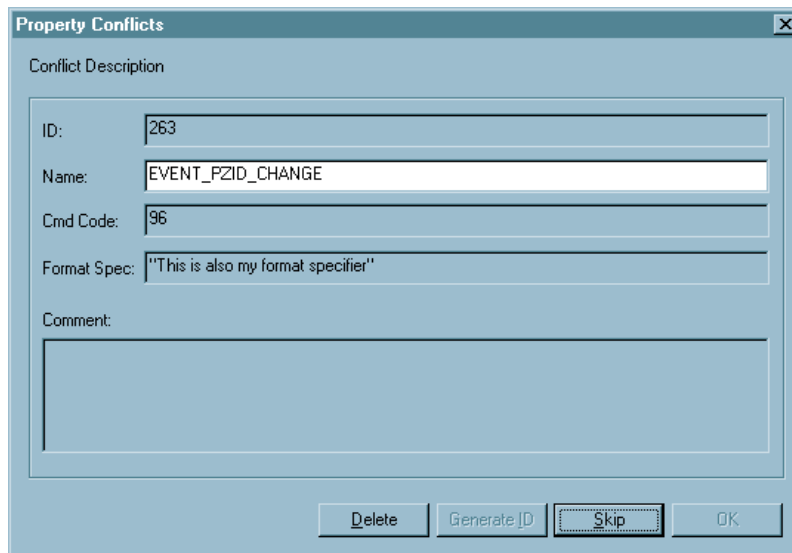


Figure 8–8 Resolving conflicting property IDs

1 If the conflict is between the names of two Properties, the Resolution dialog allows you to rename the
2 user-defined Property or delete it, as shown in Figure 8–9.



The image shows a 'Property Conflicts' dialog box with a title bar and a close button. It contains a 'Conflict Description' section with the following fields: 'ID:' with the value '263', 'Name:' with the value 'EVENT_FZID_CHANGE', 'Cmd Code:' with the value '96', and 'Format Spec:' with the value '"This is also my format specifier"'. Below these is a 'Comment:' section with a large empty text area. At the bottom of the dialog are four buttons: 'Delete', 'Generate ID', 'Skip', and 'OK'. The 'Skip' button is highlighted with a dashed border.

3
4 **Figure 8–9 Resolving conflicting property names**
5

1

9 Methods

1 Methods are functions on the subscriber that are exported and available to QXDM. Currently, all
2 functions are exported as methods. Methods are defined by the DWARF server (see Chapter 10 for
3 more information). QXDM can execute any method on the subscriber as a regular script command.
4 This feature allows you to rapidly add new script commands to QXDM without having to add new
5 diag packets and to modify the script parser.

6 To call the method, you must enter the method name and the parameters in the command window.
7 For example, to call function `foo(int x, long y)` on the subscriber, enter `@foo(1, 2)`. The
8 special character `@` preceding the method name is to tell QXDM that it is a method instead of a
9 regular script command. The return value of the method is displayed on the Command Output screen.

10
NOTE The current release of QXDM does not support methods whose parameters are pointers to structures.

1

10 DWARF Server

1 The DWARF server is the back-end that enables the Property and Method access features of QXDM.
2 It is a QXDM component that extracts DWARF2 debugging information entries from the executable
3 and linking format (ELF) file associated with a specific software build and services queries on this
4 DWARF2 database via a COM/DCOM interface. QXDM can retrieve DWARF2 debugging
5 information entries on the local machine or on a remote machine running the QXDM server
6 installation.

7 With the help of the DWARF server, QXDM allows you to access global data in the software build as
8 Properties and global functions as Methods. Addressing and type information for the Properties and
9 Methods is retrieved from the DWARF database maintained by the DWARF server.
10

11 **NOTE** Properties defined in .prop files do not require the DWARF server to function.

12 10.1 Installation

13 The DWARF database feature requires two server components, Build ID server and DWARF server.
14 As Section 2.2 explains, the server components are installed in the users' machines to act as the local
15 servers. A separate installer for the Server-Only configuration is available to establish remote servers.
16 In this installation scheme, a QXDM client has access to both local software builds and remote builds.

10.2 Build process

Modifications have been made to the build makefile to generate and register a globally unique identifier for the build. If you have an older version of the makefile, the set of rules for generating the image file must be replaced with the following:

```
@echo -----
@echo Generating a GUID for the build
@perl diag_guid_gen.pl
$(CC) $(CFLAGS) $(DMSS_CFLAGS) $(MSG_FILE) -o $(TARGET)\diag_guid.o
diag_guid.c
@echo -----
@echo Generating DIAG tables
@perl diag_table_gen.pl
$(CC) $(CFLAGS) $(DMSS_CFLAGS) $(MSG_FILE) -o $(TARGET)\diag_table.o
diag_table.c
@echo -----
@echo TARGET $@
$(LD) -$(EXETYPE) $(LFLAGS) $(OBJ_CMD) $@ $(LIBS) -VIA << -VIA <<
$(OBJECTS)
<<
$(TARGET)\diag_guid.o $(TARGET)\diag_table.o
<<
@echo Registering build GUID
@perl register_guid.pl $(TARGET).$(EXETYPE)
```

In order for a build to support properties and other diagnostic extensions, `FEATURE_DIAG_QCT_EXT` must be enabled. For method calling support, `FEATURE_DIAG_RPC` must be enabled.

During the build process, the Build ID is compiled into the output ELF file, and an entry that maps the ID to the full pathname of the ELF file is made in the registry. When QXDM initiates communication with a phone or a SURF, it asks for the Build ID before attempting any operation involving properties. The ID is passed to the build server, which locates the matching ELF file by a registry lookup. Operations involving properties are suspended until the ELF file is located and relevant information is loaded from it.

If an ELF file is moved from the build directory, it needs to be registered in order for QXDM to find it. To register an ELF file, type `reg_elf.pl <ELF filename>` in an MS-DOS shell.

10.3 Access process

To access Properties and Methods, QXDM needs the ELF file that is output by the software build process. The ELF file is associated with the software build using a unique Build ID. The Build ID is created during the modified Build Process described in the previous section. Also, the Build ID is registered with the QXDM Build Server component on the machine on which the build is performed. Figure 10–1 depicts the access process.

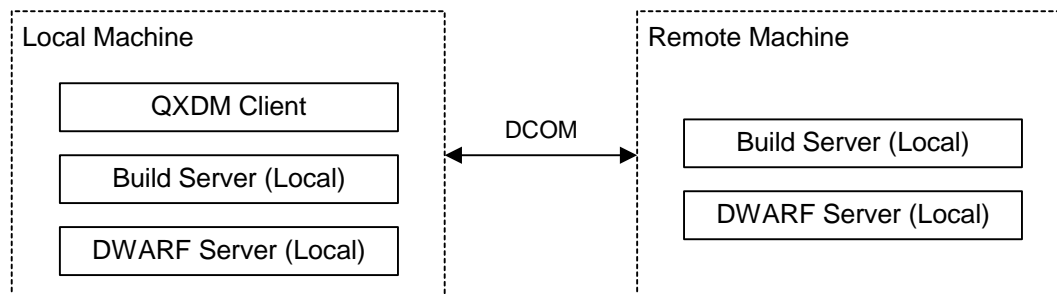


Figure 10–1 Access process

The QXDM client (GUI) looks for the Build ID when a SURF/phone is connected. If a Build ID is found in the SURF/phone, QXDM queries the Build server component on the local machine for the ELF file corresponding to the Build ID. If the Build ID is not found in the local machine, QXDM queries the Build server components in a known set of remote machines. This list of remote machines is user-configurable. The configuration process is explained in the following section.

If the Build ID was properly registered in either the local machine or one of the remote machines, QXDM invokes the DWARF server component on that machine to load the DWARF2 debugging information database and respond to queries on entities in the ELF file. The QXDM client displays the path to the ELF file in the status bar, as shown in Figure 10–2.

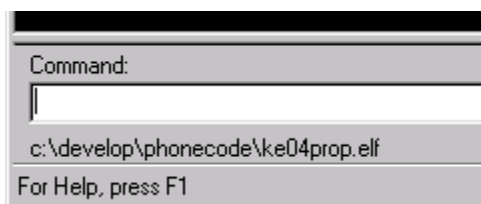
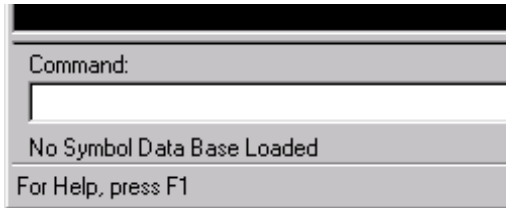


Figure 10–2 Path to the ELF file

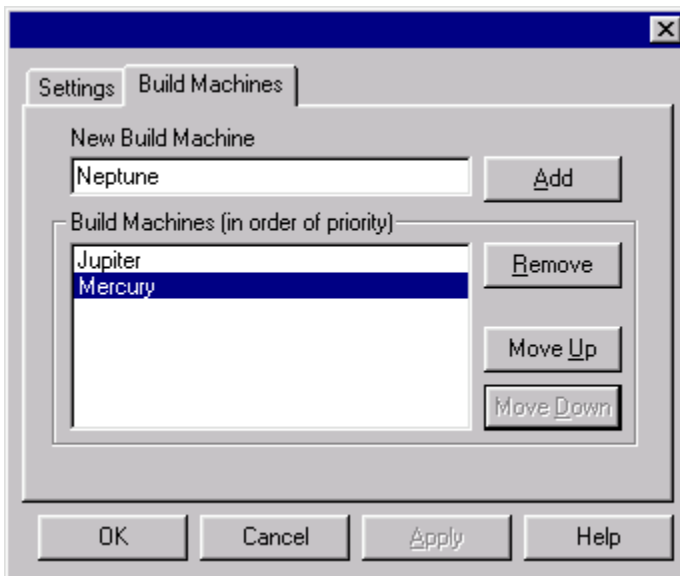
1 If the Build ID was not found in any of the specified machines, or if there are problems accessing the
2 ELF file, QXDM does not display the ELF filename in the status bar. Instead, a “No Symbol Data
3 Base Loaded” message is displayed in the status bar, as shown in Figure 10–3. In this case, QXDM
4 will not be able to support the feature of accessing Methods and Properties that are not defined in
5 .prop files.



6
7 **Figure 10–3 No Symbol Data Base Loaded message**

8 **10.4 Configuring build machines in QXDM**

9 To configure the Build Machine list in QXDM, select Options→Settings→Build Machines tab. The
10 configuration dialog, as shown in Figure 10–4, allows you to add/remove Build Machine names and
11 reorder their priority.



12
13 **Figure 10–4 Configure build machines**

11 Script Interface

1 QXDM supports a scripting API (Application Programming Interface) which uses OLE automation.
2 This interface can be used by any language that can access OLE automation objects. Examples are
3 Microsoft VBScript, Microsoft Visual Basic and Perl (needs ole.pm).

4 QXDM does not include any scripting language. Perl version 5.005_03 or newer can be installed from
5 ActiveState. Microsoft Windows Scripting host is integrated into Windows 98 and Windows 2000
6 and is available on Microsoft's website to install on other operating systems.

7 The class identifier for the QXDM object is QXDM.Application and the default interface is called
8 IQxdm. Using this information, you can load the QXDM object from your script. For Example:

9 In Perl you can load the QXDM object and its default interface by the following-

```
10 use Win32::OLE;  
11 my $qxdm_app = new Win32::OLE 'QXDM.Application';
```

12 In Microsoft VBScript you can load the QXDM object and its default interface by the following:

```
13 Dim myObject  
14 myObject = CreateObject("QXDM.Application")  
15
```

16 Once you have the QXDM object you can access all the automation interface functions directly. The
17 interface functions are divided into two categories. These are Properties and Methods.

18 The Properties are used to set or get a certain variable and can be treated like a class variable. For
19 example,

20 In Perl:

```
21 $qxdm_app->{LogMask} = "DEADBEAF";  
22
```

23 In Microsoft VBScript:

```
24 MyObject.LogMask = "DEADBEAF"  
25
```

26 The Methods are used to issue commands and are written like basic function calls. For example,

27 In Perl:

```
28 $qxdm_app->QXDMTextOut("This is a test string");  
29
```

30 In Microsoft VBScript:

```
31 MyObject.QXDMTextOut("This is a test string")  
32
```

QXDM contains 10 example Perl scripts named test00.pl through test09.pl and mashkeys.pl and 10 example Microsoft VBScript scripts named test00.vbs through test09.vbs and mashkeys.vbs. These are in the automation directory where QXDM was installed.

The following QXDM functions can be called from scripts:

Properties:

BSTR LogMask

long DataRate

long DipSwitchMask

long IsPhoneConnected

BSTR ResponsePacket

Use this property to access the response packet after calling the SendDmIcdPacket() method.

Methods:

void SendScript(BSTR Command)

void SetLoggingOn(BSTR MaskString)

void SetLoggingOff(BSTR FileName)

long ResetPhone()

long OfflineDigital()

void InjectDmIcdPacket(BSTR Packet, long Len)

void QXDMTextOut(BSTR TextString)

long SendDmIcdPacket(BSTR ReqPkt, long ReqLen, long tmo)

This function returns the response packet size. To access the response packet itself, use the ResponsePacket property.

12 QXDM Command-prompt Functions

The following additional commands can be invoked via the Command window, in addition to the regular script commands described in the Script Help display:

- @<method name> – See Chapter 9 for details.
- get_property – Read a property from the phone. See Chapter 7 for a description of properties.

Syntax: get_property <Property Name>

Example:

```
get_property version.voc_maj
version.voc_maj = 0x65
```

- print_property – Print a property definition. Refer to the Properties chapter for a description of properties.

Syntax: print_property <Property Name>

Example:

```
print_property time
# Element Name, Offset, Size, Description, Type[, Request Value]
# Property Name: time; ID: 29; Size: 9; Type: Diagnostic Packet
time.cmd_code, 0, 1, Timestamp, HEX, 0x1d
time.time_stamp, 1, 8, , HEX
time.time_stamp[0], 1, 4, , HEX
time.time_stamp[1], 4, 4, , HEX
```

- **list_properties** – List the currently supported properties or properties matching a given pattern. See Chapter 7 for a description of properties.

Syntax: **list_properties** [part of a property name]

Example:

```
list_properties s*
ser                ID: 45      DIAG type
srch2              ID: 15      Log type
srch_fing          ID: 45      Log type
state              ID: 63      DIAG type
status             ID: 12      DIAG type
sup_fer            ID: 85      DIAG type
sup_walsh_codes    ID: 86      DIAG type
svc_config         ID: 46      Log type
```

- **put_property** – Write a property to the phone. See Chapter 7 for a description of properties.

Syntax: **put_property** <Property Name> <property arguments>

Example:

```
put_property ftm_acquire 100 10 0 0 0
```

- **send_is801_message** – Send an IS-801 message to the phone. See Chapter 14 for details.

- **exit** – Exit QXDM.

- **sia_remote_control** – Start an SIA remote control session.

Syntax: **sia_remote_control** <hostname> <ESN/MIN> <repetitions>

Example:

```
sia_remote_control zeus 7605551212 0
```

- **cdm_start_log** – Start logging simple property objects to the CDM (CDMA DM).

Syntax: **cdm_start_log** <Property Name Field>

Example:

```
cdm_start_log srch_fing.pilot_off
```

- **cdm_stop_log** Stop logging simple property objects to the CDM (CDMA DM).

Syntax: **cdm_stop_log** <Property Name Field>

Example:

```
cdm_stop_log srch_fing.pilot_off
```

- **wait_event** – Wait until a requested event arrives from the phone or timeout occurs. The timeout is defined in Communication→Settings→Script Cmd Timeout.

Syntax: **wait_event** EVENT_NAME

Example:

```
wait_event EVENT_CDMA_CH_CHANGE
```


13 Temporal Analyzer

13.1 Temporal Analyzer display

To view the temporal analyzer display, as shown in Figure 13–1, select Temporal Analyzer from the middle drop-down menu on the splitter view bar (see Figure 4–6).

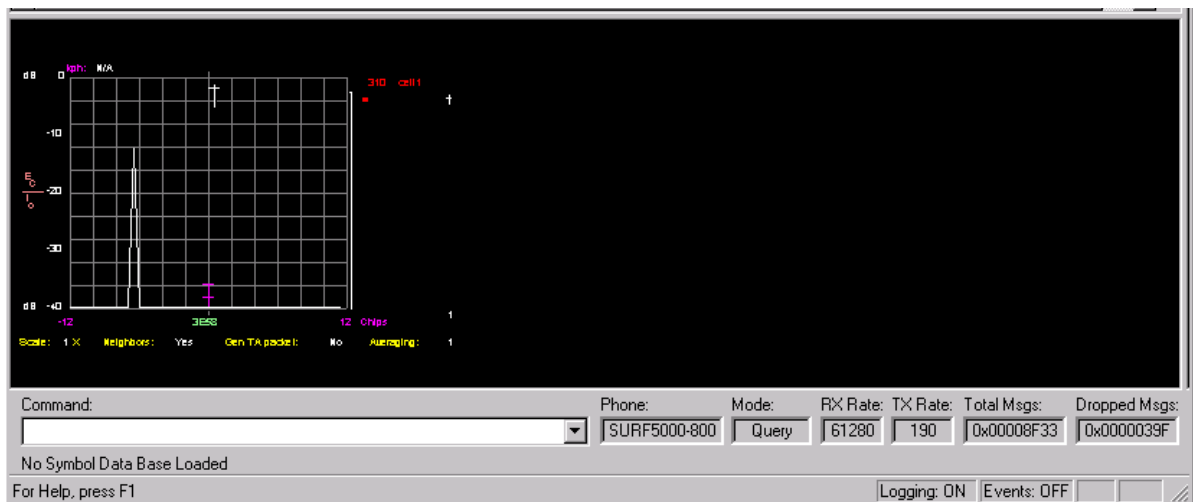


Figure 13–1 Temporal Analyzer display

For the cdma2000 1x subscriber station, the fingers that belong to the F-FCH active set and the F-SCH active set are distinguished on the temporal analyzer display by different colors.

Once the Temporal Analyzer view has been selected from the middle drop-down menu, the following graphs are accessible by either a right-click or via the Temporal Analyzer menu.

13.2 Pilot Sets display

This screen, as shown in Figure 13–2, displays the pilots belonging to the Active (A), Candidate (C), Precandidate (P), and Neighbor sets, as well as the PN offset values.

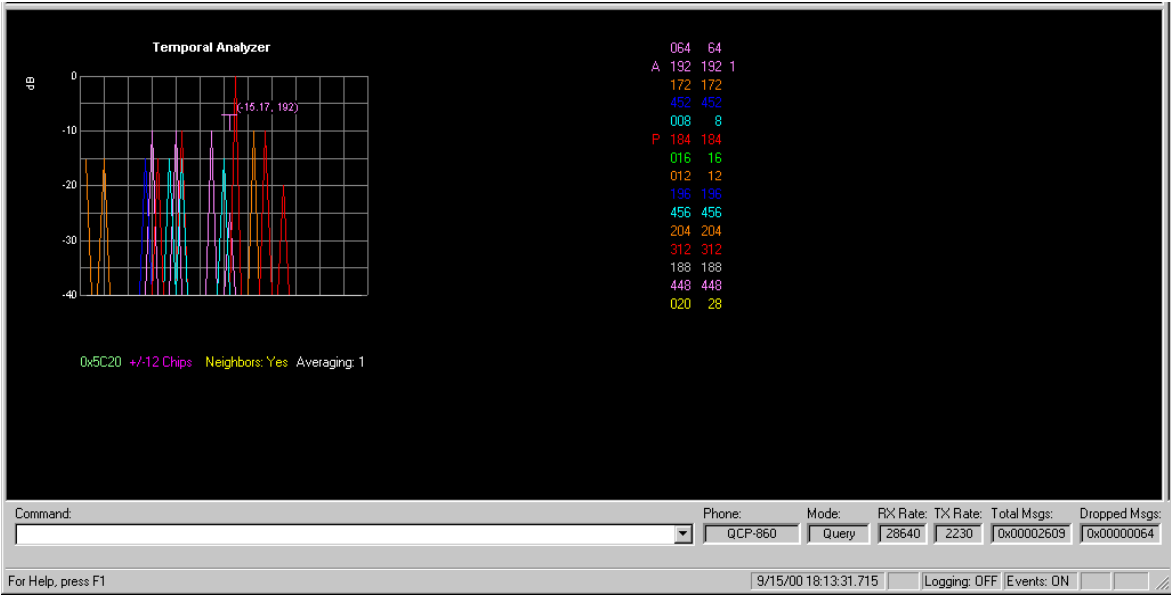


Figure 13–2 Pilot Sets display

13.3 Transmit/Receive (Tx/Rx) Adjusted Power display

This screen, as shown in Figure 13–3, displays both the received and transmitted power, as well as the amount the transmitted power is adjusted during a specified averaging period. For the cdma2000 1x subscriber station, it also displays the transmit power separately for each reverse link channel, that is, R-FCH, R-SCH0, R-SCH1, and R-DCCH.

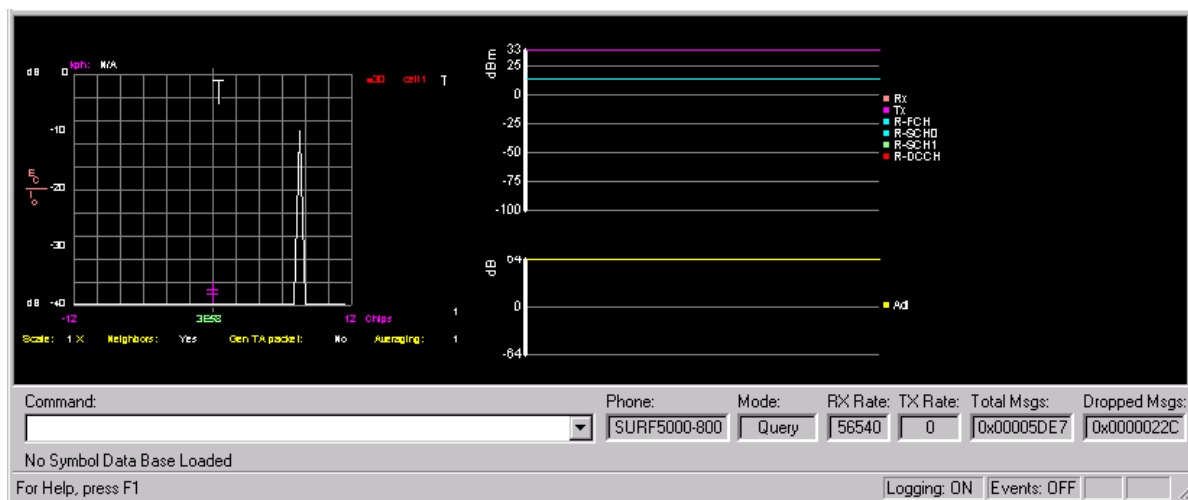


Figure 13–3 Transmit/Receive (Tx/Rx) Power display

13.4 Frame Error Rate (FER) display

The frame error rate (FER) display screen, as shown in Figure 13–4, displays the percentage of steady state errors over a specified period of time. For the cdma2000 1x subscriber station, it displays the FER for each forward link channel, that is, F-FCH, F-SCH0, F-SCH1, F-SCCH1, F-SCCH2, F-SCCH3, F-SCCH4, F-SCCH5, F-SCCH6, and F-SCCH7.

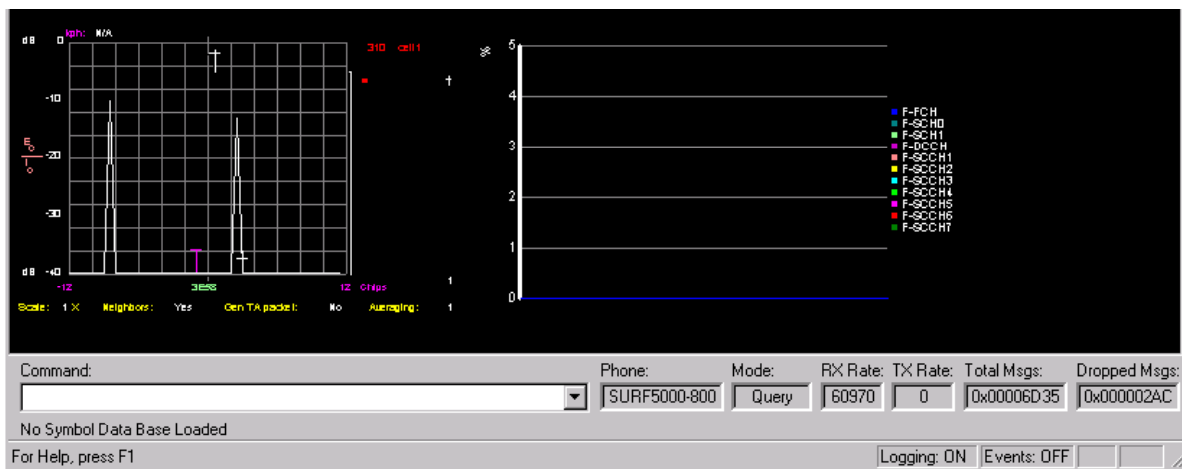


Figure 13–4 Frame Error Rate (FER) display

13.5 Frame Types (old Tx/Rx Vocoder Rate) display

This screen, as shown in Figure 13–5, displays the received and transmitted signal vocoder rate over a specified period of time. For the cdma2000 1x subscriber station, it displays the frame type rates for each forward link channel, that is, F-FCH, F-SCH0, F-SCH1, F-SCCH1, F-SCCH2, F-SCCH3, F-SCCH4, F-SCCH5, F-SCCH6, and F-SCCH7, and the frame type rates for each reverse link channel, that is, R-FCH, R-SCH0, R-SCH1, and R-DCCH.

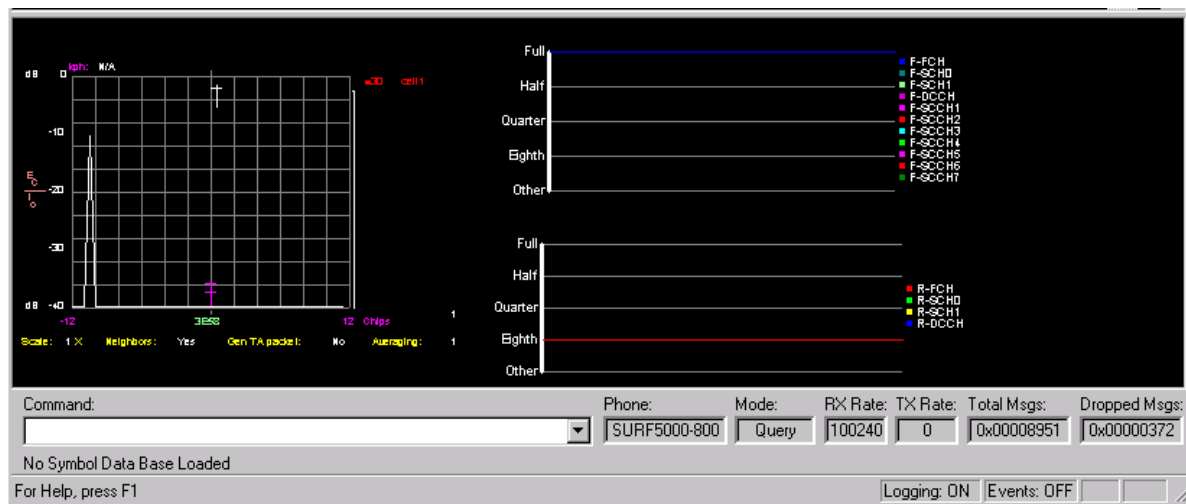


Figure 13–5 Frame Types (old Tx/Rx Vocoder Rate) display

13.7 Temporal Analyzer Configuration

You can configure the way information appears in the temporal analyzer display by using the Configuration dialog shown in Figure 13–7. You can access this dialog by either a right-click or via the Temporal Analyzer menu.

To configure the display, select various options in the Configuration dialog, then click **OK** to activate the selections.

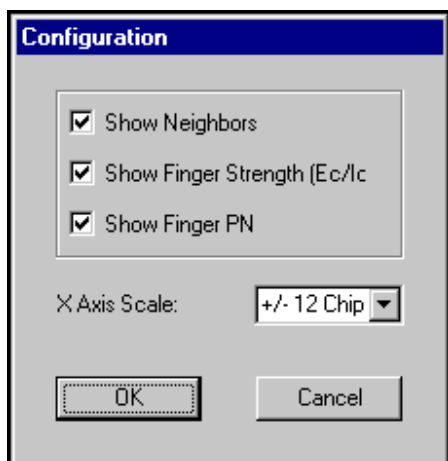


Figure 13–7 Temporal Analyzer → Configuration

1

14 Application Screens

NOTE Numerous changes have been made to this chapter. It is recommended that you review it in its entirety.

14.1 Application screens

Table 14-1 is a listing of the properties and packets utilized by QXDM's displays. Most of the screens are HTML- and JavaScript-based. While they are viewed using QXDM, they are not part of the QXDM program itself. See Section 4.3 for information on how to access these screens. Refer to the *CDMA DMSS Serial Data Interface Control Document*, 80-V1294-1, for specific descriptions of the properties and packets.

Some of the display names in Table 14-1 may have superscripted numbers following them. These numbers indicate the note information that follows the table.

Table 14-1 Properties and packets

Display name	Packet ID [decimal #]	Packets/properties used	Pkt type	Prop file
Active set information	0x1093 [147]	aset_info	Log	aset_info
	0x10C6 [198]	aset_info_ver_2	Log	aset_info
Bluetooth logs	0x103E [62]	bt_gen	Log	btLogs
	0x103F [63]	bt_bb_gen	Log	btLogs
	0x1040 [64]	bt_bb_dev	Log	btLogs
	0x1041 [65]	bt_lm	Log	btLogs
	0x1042 [66]	bt_l2	Log	btLogs
	0x1043 [67]	bt_sdp	Log	btLogs
	0x1044 [68]	bt_rc_session	Log	btLogs
	0x1045 [69]	bt_rc_dlci	Log	btLogs
	0x1046 [70]	bt_ag	Log	btLogs

Display name	Packet ID [decimal #]	Packets/properties used	Pkt type	Prop file
Factory test mode ¹	0x103A [58]	ftm_log	Log	ftm_log
Sub_cmd: (100)	0x3B [59]	ftm_acquire	Diag	ftm
(102)	0x3B [59]	ftm_assign_fwd_fch	Diag	ftm
(103)	0x3B [59]	ftm_assign_fwd_sch	Diag	ftm
(104)	0x3B [59]	ftm_assign_rev_fch	Diag	ftm
(105)	0x3B [59]	ftm_assign_rev_sch	Diag	ftm
(109)	0x3B [59]	ftm_set_mode	Diag	ftm
(110)	0x3B [59]	ftm_assign_fwd_dcch	Diag	ftm
(111)	0x3B [59]	ftm_assign_rev_dcch	Diag	ftm
Fast forward power control	0x102B [43]	ffpc	Log	ffpc
Full test data service ²	0x103D [61]	tdso_full	Log	tdso_full
	0x109A [154]	ftdso_p2	Log	ftdso_p2
	0x70 [112]	is2000_status	Diag	status2k
	0x72 [114]	reset_tdso	Diag	tdso_full
gpsOne ³	0x6C [108]	gps_sess_ctrl	Diag	gpsone
Sub_cmd: (0)	0x6C [108]	gps_sess_ctrl_start	Diag	gpsone
(1)	0x6C [108]	gps_sess_ctrl_acq	Diag	gpsone
(2)	0x6C [108]	gps_sess_ctrl_sense	Diag	gpsone
(5)	0x6C [108]	gps_sess_ctrl_pl	Diag	gpsone
(8)	0x6C [108]	gps_sess_ctrl_almanac	Diag	gpsone
(9)	0x6C [108]	gps_sess_ctrl_ephemeris	Diag	gpsone
	0x6D [109]	gps_search_grid	Diag	gpsone
GPS statistics ⁴	0x1038 [56]	gps_stats	Log	gps_stats
	0x6E [110]	gps_statistics	Diag	gpsone
HDR				
■ Air link summary	0x1068 [104]		Log	
	0x1069 [105]		Log	
■ Fingers	0x1064 [100]	hdr_finger_data	Log	hdrlog
■ Fwd statistics	0x105F [95]	hdr_asp	Log	hdrlog
	0x1061 [97]	hdr_drc_arq_buffer	Log	hdrlog
	0x1064 [100]	hdr_finger_data	Log	hdrlog
	0x1067 [103]	hdr_ftc_rate_count	Log	hdrlog
	0x106F [111]	hdr_page_msg	Log	hdrlog
	0x107F [127]	hdr_fwd_rate_stat	Log	hdrlog
Subsys id/cmd: (5/4)	0x4B [75]	hdr_reset_page_msg_cnt	Diag	hdrlog
(5/1)	0x4B [75]	hdr_reset_fwd_stats	Diag	hdrlog
■ Pilot sets	0x107A [122]	hdr_pilot_sets	Log	hdrlog
	0x1064 [100]	hdr_finger_data	Log	hdrlog

Display name	Packet ID [decimal #]	Packets/properties used	Pkt type	Prop file
■ Protocol summary	0x107E [126]	hdr_state_info	Log	hdrlog
	0x106B [107]	hdr_session_attempt	Log	hdrlog
	0x106E [110]	hdr_conn_attempt	Log	hdrlog
Subsys id/cmd: (5/5)	0x4B [75]	hdr_reset_session_atmpt_cnt	Diag	hdrlog
(5/2)	0x4B [75]	hdr_reset_conn_atmpt_cnt	Diag	hdrlog
■ Rev statistics	0x1060 [96]	hdr_rl_pkt_summary	Log	hdrlog
	0x1062 [98]	hdr_rtc_rate_cnt	Log	hdrlog
	0x106C [108]	hdr_access_attempt	Log	hdrlog
	0x107E [126]	hdr_state_info	Log	hdrlog
Subsys id/cmd: (5/2)	0x4B [75]	hdr_reset_rev_stats	Diag	hdrlog
(5/0)	0x4B [75]	hdr_reset_access_atmpt_cnt	Diag	hdrlog
■ RLP statistics	0x1070 [112]	hdr_rlp_stats	Log	hdrlog
Subsys id/cmd: (5/6)	0x4B [75]	hdr_reset_rlp_stats	Diag	hdrlog
■ Status	0x105F [95]	hdr_asp	Log	hdrlog
	0x1062 [98]	hdr_rtc_rate_cnt	Log	hdrlog
	0x1064 [100]	hdr_finger_data	Log	hdrlog
	0x107A [122]	hdr_pilot_sets	Log	hdrlog
	0x107E [126]	hdr_state_info	Log	hdrlog
	0x107F [127]	hdr_fwd_rate_stat	Log	hdrlog
	0x1080 [128]	hdr_sector_info	Log	hdrlog
Subsys id/cmd: (5/7)	0x4B [75]	hdr_at_config	Diag	hdrlog
	0xC [12]	status	Diag	status
■ Temporal analyzer	0x107D [125]	hdr_gen_ta	Log	hdr_gen_ta
Log priorities				
Sub_cmd ⁵ : (2)	0x61 [97]	log_priorities_req	Diag	logPriorities
(2)	0x61 [97]	log_priorities_rsp	Diag	logPriorities
Markov ⁶	0x1A [26]	markov	Diag	markov
	0x1B [27]	DIAG_MARKOV_RESET_F	Diag	
MDR statistics	0x56 [86]	sup_walsh_codes	Diag	sup_walsh_codes
MSP				
Subsys id/cmd: (1/1)	0x4B [75]	DIAG_ZREX_SET_LOG_MASK_F	Diag	
(1/3)	0x4B [75]	DIAG_ZREX_DRIVER_INFO_F	Diag	
(1/4)	0x4B [75]	DIAG_ZREX_SW_VERSIONS_F	Diag	
(1/5)	0x4B [75]	DIAG_ZREX_TOUCHSCREEN_EMU_F	Diag	
(1/6)	0x4B [75]	DIAG_ZREX_LAUNCH_ZX_F	Diag	

Display name	Packet ID [decimal #]	Packets/properties used	Pkt type	Prop file
MUX statistics ²	0x70 [112]	is2000_status	Diag	status2k
Sub_cmd: (1)	0x62 [98]	diag_parm_retrieve_is2k_std	Diag	parm_retrieve
(3)	0x62 [98]	diag_parm_retrieve_is2k_ext	Diag	parm_retrieve
(1)	0x62 [98]	diag_parm_retrieve_dcch	Diag	parm_retrieve
(1)	0x62 [98]	diag_parm_retrieve_lower	Diag	parm_retrieve
MUX1 trfc & Layer2 stats	0x43 [67]	IS95Aparam	Diag	IS95Aparam
MUX2 trfc & sec stats ⁷	0x43 [67]	IS95Aparam	Diag	IS95Aparam
MUX2 trfc ch stats ⁸	0x43 [67]	IS95Aparam	Diag	IS95Aparam
NV items ⁹		See prop file for complete list.	Diag	nv_items
Paging & access stats ¹⁰	0x43 [67]	IS95Aparam	Diag	IS95Aparam
Quick paging channel	0x1030 [48]	qpch	Log	qpch
Register peek/poke ¹¹	0x0 [0]	version	Diag	version
		See prop file for complete list.	Abs Addr ¹²	msm3000_registers and msm5000_registers
Retrievable parameters				
Sub_cmd: (3)	0x62 [98]	diag_parm_retrieve_is2k_ext	Diag	parm_retrieve
(1)	0x62 [98]	diag_parm_retrieve_is2k_std	Diag	parm_retrieve
(1)	0x62 [98]	diag_parm_retrieve_lower	Diag	parm_retrieve
(1)	0x62 [98]	diag_parm_retrieve_upper	Diag	parm_retrieve
(0)	0x62 [98]	diag_parm_retrieve_range_std	Diag	parm_retrieve
(2)	0x62 [98]	diag_parm_retrieve_range_ext	Diag	parm_retrieve
(1)	0x62 [98]	diag_parm_retrieve_dcch	Diag	parm_retrieve
RLP3 statistics logging ²	0x1031 [49]	rlpstats	Log	rlpstats
	0x71 [113]	reset_rlpstats	Diag	rlpstats
	0x70 [112]	is2000_status	Diag	status2k
SBI data access	0x75 [117]	DIAG_SBI_READ_F	Diag	
	0xE4 [228]	DIAG_SBI_WRITE_F	Diag	
Script help ¹⁴		Uses no properties		
Searcher & finger information	0x102D [45]	srch_fing	Log	srch_fing
Security services SSL sessions	0x10A9 [169]	ssl_session_info	Log	QXDM database
	0x10AA[170]	ssl_session_info	Log	QXDM database
	0x10AB[171]	ssl_session_info	Log	QXDM database
	0x10AC[172]	ssl_session_stats	Log	QXDM database
	0x10AD[173]	ssl_tx_stats	Log	QXDM database
	0x10AE[174]	ssl_rx_stats	Log	QXDM database

Display name	Packet ID [decimal #]	Packets/properties used	Pkt type	Prop file
Service configuration info	0x10C5 [197]	svc_config_ver_2	Log	svc_config
	0x102E [46]	svc_config	Log	svc_config
Simple test data service ²	0x1032 [50]	tdso	Log	tdso
	0x1099 [153]	stdso_p2	Log	stdso_p2
	0x70 [112]	is2000_status	Diag	status2k
	0x72 [114]	reset_tdso	Diag	tdso_full
Status	0x70 [112]	is2000_status	Diag	status2k
Subsys id/cmd: (4/0)	0x4B [75]	wcdma_version	Diag	wcdma_version
	0x0 [0]	version	Diag	version
	0xC [12]	status	Diag	status
	0x3F [63]	state	Diag	state
	0x1019 [25]	genta	Log	genta
Temporal analyzer	0x100B [11]	LOG_FING_C	Log	
	0x100C [12]	LOG_SRCH_C	Log	
	0x100F [15]	LOG_SRCH2_C	Log	
	0x1013 [19]	LOG_ACC_INFO_C	Log	
	0x1019 [25]	LOG_GENRL_TA_C	Log	
	0x1029 [41]	LOG_FWD_FRAME_TYPES_C	Log	
	0x102A [42]	LOG_REV_FRAME_TYPES_C	Log	
	0x102C [44]	LOG_REV_PCTRL_C	Log	
	0x102D [45]	LOG_SRCH_FING_INFO_C	Log	
	0x1031 [49]	LOG_RLP_STAT_C	Log	
WCDMA¹⁵				
■ AGC ¹⁶	0x4105 [261]	WCDMA_AGC_C	Log	
■ Block error rate	0x4116 [278]	dl_trch_bler	Log	wcdma_l1
■ Layer1 state	0x4104 [260]	wcdma_finger_ta	Log	wcdma_Finger
	0x410A [266]	common_dl_phychan	Log	achannels
	0x410C [268]	ul_dpch	Log	achannels
	0x4110 [272]	aset	Log	achannels
	0x4111 [273]	nset	Log	achannels
	0x412E [302]	set_ref_pos	Log	achannels
■ MAC channel mapping	0x411D [285]	wcdma_dl_mac_cfg	Log	maclog
	0x4117 [279]	wcdma_ul_mac_cfg	Log	maclog
■ MAC parameters	0x4123 [291]	mac_rach_cfg	Log	maclog
	0x4117 [279]	wcdma_ul_mac_cfg	Log	maclog
	0x411D [285]	wcdma_dl_mac_cfg	Log	maclog

Display name	Packet ID [decimal #]	Packets/properties used	Pkt type	Prop file
■ NAS CS connection mgmt	0x4137 [311]	wcdma_nas_cs_connection_bc	Log	wcdma_l4
	0x4133 [307]	wcdma_cs_call_info	Log	wcdma_l4
■ NAS errors	0x413C [316]	wcdma_nas_error_message	Log	wcdma_l4
■ NAS mobility mgmt	0x4131 [305]	wcdma_nas_mm_state	Log	wcdma_l4
	0x4130 [304]	wcdma_nas_gmm_state	Log	wcdma_l4
	0x4132 [306]	wcdma_nas_reg_state	Log	wcdma_l4
	0x4135 [309]	wcdma_mm_info	Log	wcdma_l4
	0x4138 [312]	wcdma_nas_ue_dynamic_id	Log	wcdma_l4
	0x4139 [313]	wcdma_nas_ue_static_id	Log	wcdma_l4
■ NAS PS connection mgmt	0x4136 [310]	wcdma_nas_ps_connection_qos	Log	wcdma_l4
	0x4134 [308]	wcdma_ps_call_info	Log	wcdma_l4
■ Physical channels	0x410C [268]	ul_dpch	Log	achannels
	0x410D [269]	ul_phychan_prach	Log	wcdma_prach
	0x410A [266]	common_dl_phychan	Log	achannels
	0x410B [267]	ded_dl_phychan	Log	achannels
■ RLC DL AM channel parameters	0x4122 [290]	wcdma_rlc_dl_am	Log	wcdma_rlc_layer
	0x411E [286]	wcdma_rlc_dl_cipherring	Log	wcdma_rlc_layer
■ RLC DL TM channel parameters	0x4120 [288]	wcdma_rlc_dl_tm	Log	wcdma_rlc_layer
■ RLC DL UM channel parameters	0x4121 [289]	wcdma_rlc_dl_um	Log	wcdma_rlc_layer
■ RLC Log channel setup UL/DL	0x4117 [279]	wcdma_ul_mac_cfg	Log	maclog
	0x4119 [281]	wcdma_rlc_ul_state	Log	wcdma_rlc_layer
	0x411D [285]	wcdma_dl_mac_cfg	Log	maclog
	0x411F [287]	wcdma_rlc_dl_state	Log	wcdma_rlc_layer
■ RLC UL AM channel parameters	0x411C [284]	wcdma_rlc_ul_am	Log	wcdma_rlc_layer
	0x4118 [280]	wcdma_rlc_ul_cipherring	Log	wcdma_rlc_layer
■ RLC UL TM channel parameters	0x411A [282]	wcdma_rlc_ul_tm	Log	wcdma_rlc_layer
■ RLC UL UM channel parameters	0x411B [283]	wcdma_rlc_ul_um	Log	wcdma_rlc_layer
■ RRC status	0x4126 [294]	wcdma_rrc_protocol_errors	Log	wcdma_rrc
	0x412B [299]	wcdma_sib	Log	wcdma_rrc
	0x4127 [295]	wcdma_cell_id	Log	wcdma_rrc
	0x4125 [293]	wcdma_rrc_modes_and_states	Log	wcdma_rrc
■ Temporal analyzer	0x4104 [260]	WCDMA_FINGER_TA_INFO_C	Log	wcdma_Finger
■ TFCS downlink	0x4115 [277]	dl_tfcs	Log	wcdma_l1

Display name	Packet ID [decimal #]	Packets/properties used	Pkt type	Prop file
■ TFCS uplink	0x4124 [292]	wcdma_mac_ul_tfcs_cfg	Log	maclog
	0x4117 [279]	wcdma_ul_mac_cfg	Log	maclog
■ Transport channels	0x4108 [264]	dl_cctch	Log	wcdma_dl_cctrch
	0x4109 [265]	ul_cctrch	Log	wcdma_ul_cctrch

The following sections are note information for Table 14-1.

14.1.1 Note 1

See *Factory Test Mode Application Note*, CL93-V1974-1, for specific descriptions

14.1.2 Note 2

The combo box will dynamically show which rate set is being used if the Autodetect box is checked. If no rate set can be detected, the combo box will contain Rate Set Unknown. If the user unchecks the Autodetect box and sets the combo box to a particular rate set, the values will only refresh if the selected rate is in use or if it is unknown.

14.1.3 Note 3

gpsOne – A new command, `send_is801_message`, can be used on the QXDM command line to send IS-801 messages to the phone. The syntax for the command is as follows (square brackets indicate an optional parameter):

```
send_is801_message <GpsControl> <FileName> [<SetBitOption>]
```

<GpsControl> defines messages for acquisition assistance, sensitivity assistance, and position location response.

<FileName> is the name of a file containing text data in the struct iteration language toolkit (SILK) input format, describing the IS-801 message fields. Example input file templates are located in the QXDM\reference folder of the QXDM installation.

CAUTION The templates have values that work with SILK. Providing incorrect values may cause QXDM to hang while the SILK functions wait for additional data.

Include the <SetBitOption> parameter to give the mobile additional information.

The values for the <GpsControl>, <FileName>, and (optional) <SetBitOption> parameters must be grouped as shown in Table 14-2.

Table 14-2 Required syntax for send_is801_message

IS-801 message	Command
	send_is801_message <GpsControl> <FileName> <SetBitOption>
Acquisition assistance	send_is801_message 1 silkspecAA.txt UseMessageTime
Sensitivity assistance	send_is801_message 2 silkspecSA.txt WipeOff
Provide position location response	send_is801_message 5 silkspecPL.txt (Reserved)
Provide Almanac data	send_is801_message 8 silkspecAlmanac.txt (Reserved)
Provide Ephemeris data	send_is801_message 9 silkspecEphemeris.txt (Reserved)

The gpsOne screen displays the gpsOne commands. Refer to the *CDMA DMSS Serial Data Interface Control Document*, 80-V1294-1, for diag message descriptions. Refer to the *gpsOne Position Determination Messaging and Parameters* document for a description of the IS-801 message fields.

14.1.4 Note 4

GPS statistics – Refer to the *CDMA DMSS Serial Data Interface Control Document*, 80-V1294-1, for specific GPS Statistics log record descriptions. In addition to the fields, the following calculations are displayed:

- % Successful Sessions (Yield) = NUM_SUCCESSFUL_SESSIONS/NUM_SESSIONS
- Current Lat Error = TRUE_LAT – LAT
- Current Long Error = TRUE_LONG – LONG
- Current Height Error = TRUE_HEIGHT – HEIGHT
- Current Error = $\sqrt{LAT_error^2 + LONG_error^2}$
- $$BIAS = \sqrt{\left(\frac{\sum LAT_error}{NUM_SUCCESSFUL_SESSIONS}\right)^2 + \left(\frac{\sum LONG_error}{NUM_SUCCESSFUL_SESSIONS}\right)^2}$$
- NUM_SUCCESSFUL_SESSIONS = Sum of SUCCESS_TYPE = PD_SUCCESS
- 68% cdf accuracy = calculated numerically by ordering the successful samples by error, and taking the 68th percentile
- 95% cdf accuracy = calculated numerically by ordering the successful samples by error, and taking the 95th percentile
- TOTAL_TIME_TO_FIX = sum of TIME_TO_FIX
- Average time to fix = TOTAL_TIME_TO_FIX/NUM_SUCCESSFUL_SESSIONS
- NUM_SESSIONS = sum of session log records

- Sessions per AA = $\text{NUM_SESSIONS} / \text{NUM_AA_GETS}$
- Sessions per SA = $\text{NUM_SESSIONS} / \text{NUM_SA_GETS}$
- Average tune away time = $\text{TIME_TUNE_AWAY} / \text{NUM_TUNE_AWAY}$
- % tune away time < 100ms = $\text{NUM_100MS} * 100 / \text{NUM_SUCCESSFUL_SESSIONS}$
- % tune away time < 500ms = $\text{NUM_400MS} * 100 / \text{NUM_SUCCESSFUL_SESSIONS}$
- % tune away time < 2100ms = $\text{NUM_1600MS} * 100 / \text{NUM_SUCCESSFUL_SESSIONS}$

14.1.5 Note 5

The Log Priorities display is used to change the priority at which messages and logs are sent from a streaming phone by setting different priority values.

14.1.6 Note 6

The Markov Statistics screen displays statistics generated while making a Markov call. Clicking **Reset** will reset the Markov values on the phone.

14.1.7 Note 7

The MUX2 Traffic Channel and Secondary Statistics screen displays the MUX2 traffic channel and secondary statistics when making a call. Clicking **Reset** will reset the values on the phone.

14.1.8 Note 8

The MUX2 Traffic Channel Statistics screen displays the MUX2 traffic channel statistics when making a call. Clicking **Reset** will reset the values on the phone.

14.1.9 Note 9

The NV Items display allows you to read and write NV items. NV items are grouped by categories. When you select a category, only the NV items within that category are displayed.

When you click an NV item, it will expand to display the names and values of all fields within the item. Click **Read** to reread the current value of the selected NV item. After modifying the values, click **Write** to write the updated values back to the phone.

14.1.10 Note 10

The Paging and Access Statistics screen displays the paging channel and access channel statistics. Clicking **Reset** will reset the values on the phone.

14.1.11 Note 11

Register peek/poke provides a convenient way for developers to peek and poke into the MSM registers. This provides functionality similar to the memory peek/poke (see Section 4.4.5 for more information). By selecting the register name, you can observe the register contents at run-time instead of having to enter the register address. You can also modify the register contents and observe the effect on the phone operation. Each register can have an associated mask. In addition to viewing and editing, you can specify whether the mask is applied when changing the register contents.

The MSM register set is automatically loaded according to the mobile model when QXDM is connected to the FFA or SURF. Note that the screen is not updated continuously. You must click **Refresh** to refresh the values. Figure 14–1 illustrates the register peek/poke view.

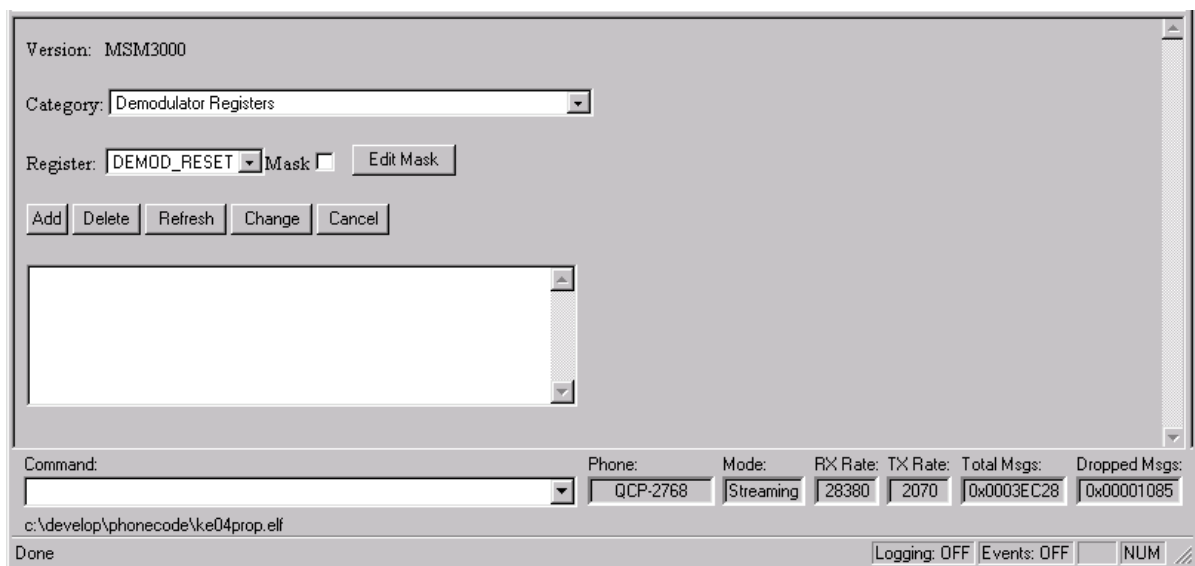


Figure 14–1 Register Peek/Poke Viewer

Category

Lists the register functional groupings.

Register

Lists the MSM registers within the selected category.

Mask

This check box indicates whether or not the mask is applied when modifying the register contents.

Edit mask

Used to view and edit the mask associated with the selected register in a Mask dialog, as shown in Figure 14–2.

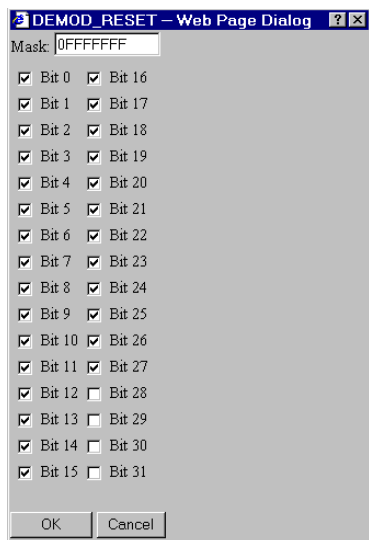


Figure 14–2 Mask dialog

Add

Adds the selected register to the edit screen to view the register contents.

Delete

Deletes the last added register.

Refresh

Refreshes all selected register contents.

Change

Used to apply the change after editing the register contents. You can also press ENTER to apply the change.

Cancel

Used to discard the change. You can also press CANCEL to discard the change.

14.1.12 Note 12

These are Absolute Address properties. Refer to section 6.3 for a description of this property type.

14.1.13 Note 13

The Retrievable Parameters screen displays the standard and nonstandard retrievable parameters selected. Parameters to be viewed can be added or removed by clicking **Select**.

14.1.14 Note 14

The Script Help screen contains help on all DM script commands. See Chapter 11 for more information on how to execute DM script commands.

14.1.15 Note 15

Refer to the *Serial Interface Control Document for W-CDMA*, 80-V2708-1, for specific descriptions of the following WCDMA properties and packets.

14.1.16 Note 16

This WCDMA AGC screen displays both the received and transmitted power, as well as the amount the transmitted power is adjusted during a specified averaging period.

15 Troubleshooting

This chapter describes several situations you might encounter while using QXDM and solutions to try.

Problem The license file is missing, invalid, or has expired.

Solution Send email to asicapps@qualcomm.com to extend your QXDM license, repair the license file, or to order CAIT.

Problem QXDM frequently disconnects and reconnects to the phone.

Solution Increase the RX Timeout. See Chapter 3 for more information.

Problem QXDM takes too long to determine that the phone is disconnected.

Solution Decrease the RX Timeout. See Chapter 3 for more information.

Problem QXDM times out when sending script commands.

Solution Increase the Script Cmd Timeout. See Chapter 3 for more information.

Problem QXDM is sending data too quickly to the target.

Solution Increase the RXTX Throttle Timeout. See Chapter 3 for more information.

Problem Logging-related data is not showing up on screens that use logging data such as Status or Temporal Analyzer.

Solution Verify that logging is turned on. See Chapter 5 for more information.

NOTE For problems not covered here, please send email to asicapps@qualcomm.com.

1