MICROMUSE
NETCOOL® SOLUTIONS

Netcool® for Wireless User Quality™

2.0

Monitoring Unit Guide

Micromuse Disclaimer of Warranty and Statement of Limited Liability

Micromuse provides this document "as is", without warranty of any kind, either express or implied, including, but not limited to, the implied warranties of merchantability, fitness for a particular purpose or non-infringement. This document may contain technical inaccuracies or typographical errors. Micromuse may make improvements and changes to the programs described in this document or this document at any time without notice. Micromuse assumes no responsibility for the use of the programs or this document except as expressly set forth in the applicable Micromuse agreement(s) and subject to terms and conditions set forth therein. Micromuse does not warrant that the functions contained in the programs will meet your requirements, or that the operation of the programs will be uninterrupted or error-free. Micromuse shall not be liable for any indirect, consequential or incidental damages arising out of the use or the ability to use the programs or this document.

Micromuse specifically disclaims any express or implied warranty of fitness for high risk activities.

Micromuse programs and this document are not certified for fault tolerance, and are not designed, manufactured or intended for use or resale as on-line control equipment in hazardous environments requiring fail-safe performance, such as in the operation of nuclear facilities, aircraft navigation or communication systems, air traffic control, direct life support machines, or weapons systems ("High Risk Activities") in which the failure of programs could lead directly to death, personal injury, or severe physical or environmental damage.

Compliance with Applicable Laws; Export Control Laws

Use of Micromuse programs and documents is governed by all applicable federal, state and local laws. All information therein is subject to U.S. export control laws and may also be subject to the laws of the country where you reside.

All Micromuse programs and documents are commercial in nature. Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in DFARS 252.227-7015 and FAR 52.227-19.

Trademarks and Acknowledgements

Micromuse and Netcool are registered trademarks of Micromuse.

Other Micromuse trademarks include but are not limited to: Netcool/OMNIbus, Netcool/OMNIbus for Voice Networks, Netcool/Reporter, Netcool/Internet Service Monitors, Netcool/ISM, Netcool/ISM Global Perspective, Netcool/NT Service Monitors, Netcool/Wireless Service Monitors, Netcool/WSM, Netcool/Usage Service Monitors, Netcool/USM, Netcool/Telco Service Monitors, Netcool/TSM, Netcool/Fusion, Netcool/Data Center Monitors, Netcool DCM, Netcool/Impact, Netcool/Visionary, Netcool/Precision, Netcool Probes & Monitors, Netcool Desktops, Netcool Gateways, Netcool Impact/Data Source Adaptors, Netcool EventList, Netcool Map, Netcool Virtual Operator, Netcool/Precision for IP Networks, Netcool/Precision for Transmission Networks, Netcool/Firewall, Netcool/Wave, Netcool/Webtop, Netcool TopoViz, Netcool/SM Operations, Netcool/SM Configuration, Netcool/OpCenter, Netcool/System Service Monitors, Netcool/SSM, Netcool/Application Service Monitors, Netcool/ASM, Netcool/ISM WAM, Netcool/SM Reporter, Netcool for Asset Management, Netcool/Realtime Active Dashboards, Netcool/Dashboards, Netcool/RAD, Netcool for Voice over IP, Netcool for Security Management, Netcool Security Manager, Netcool/Portal 2.0 Premium Edition, Netcool ObjectServer, Netcool/RAD, Netcool GUI Foundation, Netcool Installer, Netcool Licensing, Netcool/Software Developers Kit, NGF, Micromuse Alliance Program, Micromuse Channel Partner, Authorized Netcool Reseller, Netcool Ready, Netcool Solutions, Netcool Certified, Netcool Certified Consultant, Netcool Certified Trainer, Netcool CCAI Methodology, Micromuse University, Microcorrelation, Acronym, Micromuse Design, Integration Module for Netcool, The Netcool Company, VISIONETCOOL, Network Slice.

Micromuse acknowledges the use of I/O Concepts Inc. X-Direct 3270 terminal emulators and hardware components and documentation in Netcool/Fusion. X-Direct ©1989-1999 I/O Concepts Inc. X-Direct and Win-Direct are trademarks of I/O Concepts Inc.

Netcool/Fusion contains IBM Runtime Environment for AIX®, Java™ Technology Edition Runtime Modules © Copyright IBM Corporation 1999. All rights reserved.

Netcool/Precision IP includes software developed by the University of California, Berkeley and its contributors.

Micromuse acknowledges the use of MySQL in Netcool/Precision for IP Networks. Copyright © 1995, 1996 TcX AB & Monty Program KB & Detron HB Stockholm SWEDEN, Helsingfors FINLAND and Uppsala SWEDEN. All rights reserved.

Micromuse acknowledges the use of the UCD SNMP Library in Netcool/ISM and the Netcool/OMNIbus SNMP Writer Gateway. Copyright © 1989, 1991, 1992 by Carnegie Mellon University. Derivative Work - Copyright © 1996, 1998, 1999, 2000 The Regents of the University of California. All rights reserved.

Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU and The Regents of the University of California not be used in advertising or publicity pertaining to distribution of the software without specific written permission.

CMU AND THE REGENTS OF THE UNIVERSITY OF CALIFORNIA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL CMU OR THE REGENTS OF THE UNIVERSITY OF CALIFORNIA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM THE LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Portions of the Netcool/OMNIbus code are copyright (C) 1989-95 GROUPE BULL.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL GROUPE BULL BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Portions of the Netcool/ISM code are copyright ©2001, Cambridge Broadband Ltd. All rights reserved.

Portions of the Netcool/ISM code are copyright © 2001, Networks Associates Technology, Inc. All rights reserved.

Micromuse acknowledges the use of Viador Inc. software and documentation for Netcool/Reporter. Viador © 1997-1999 is a trademark of Viador Inc.

PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Sleepycat software is available from
http://downloads.sleepycat.com/db-3.0.55.zip.

Portions of Netcool/SM Reporter are copyright (c) 1990, 1993, 1994, 1995. The Regents of the University of California. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE REGENTS AND CONTRIBUTORS ``AS IS'' AND* ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE REGENTS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Portions of Netcool/SM Reporter are copyright (c) 1995, 1996. The President and Fellows of Harvard University. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY HARVARD AND ITS CONTRIBUTORS ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL HARVARD OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Netcool/SM Reporter includes the Jetty Package which is copyright (c) 1998 Mort Bay Consulting Pty. Ltd. (Australia) and others. Individual files in this package may contain additional copyright notices. The javax.servlet packages are copyright Sun Microsystems Inc.

1. The Standard Version of the Jetty package is available from http://www.mortbay.com.

2. You may make and distribute verbatim copies of the source form of the Standard Version of this Package without restriction, provided that you include this license and all of the original copyright notices and associated disclaimers.

3. You may make and distribute verbatim copies of the compiled form of the Standard Version of this Package without restriction, provided that you include this license.

4. You may apply bug fixes, portability fixes and other modifications derived from the Public Domain or from the Copyright Holder. A Package modified in such a way shall still be considered the Standard Version.

5. You may otherwise modify your copy of this Package in any way, provided that you insert a prominent notice in each changed file stating how and when you changed that file, and provided that you do at least ONE of the following:

a) Place your modifications in the Public Domain or otherwise make them Freely Available, such as by posting said modifications to Usenet or an equivalent medium, or placing the modifications on a major archive site such as ftp.uu.net, or by allowing the Copyright Holder to include your modifications in the Standard Version of the Package.

b) Use the modified Package only within your corporation or organization.

c) Rename any non-standard classes so the names do not conflict with standard classes, which must also be provided, and provide a separate manual page for each non-standard class that clearly documents how it differs from the Standard Version.

d) Make other arrangements with the Copyright Holder.

6. You may distribute modifications or subsets of this Package in source code or compiled form, provided that you do at least ONE of the following:

a) Distribute this license and all original copyright messages, together with instructions (in the manual page or equivalent) on where to get the complete Standard Version.

b) Accompany the distribution with the machine-readable source of the Package with your modifications. The modified package must include this license and all of the original copyright notices and associated disclaimers, together with instructions on where to get the complete Standard Version.

c) Make other arrangements with the Copyright Holder.

7. You may charge a reasonable copying fee for any distribution of this Package. You may charge any fee you choose for support of this Package. You may not charge a fee for this Package itself. However, you may distribute this Package in aggregate with other (possibly commercial) programs as part of a larger (possibly commercial) software distribution provided that you meet the other distribution requirements of this license.

8. Input to or the output produced from the programs of this Package do not automatically fall under the copyright of this Package, but belong to whomever generated them, and may be sold commercially, and may be aggregated with this Package.

9. Any program subroutines supplied by you and linked into this Package shall not be considered part of this Package.

10. The name of the Copyright Holder may not be used to endorse or promote products derived from this software without specific prior written permission.

11. This license may change with each release of a Standard Version of the Package. You may choose to use the license associated with version you are using or the license of the latest Standard Version.

12. THIS PACKAGE IS PROVIDED “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Netcool/SM Reporter includes FreeMarker, a tool that allows Java programs to use

templates to generate HTML or other text output that contains dynamic content. Copyright (C) 1998, 2002 Benjamin Geer. E-mail: beroul@users.sourceforge.net

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

Neither the name "Freemarker" nor any of the names of the project contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Copyright 2006 Micromuse. Portions of Netcool/WSM are licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0. Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

Micromuse acknowledges the use of Digital X11 in Netcool/Precision for IP Networks. Copyright 1987, 1988 by Digital Equipment Corporation, Maynard, Massachusetts, All Rights Reserved. DIGITAL DISCLAIMS ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS, IN NO EVENT SHALL DIGITAL BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Micromuse acknowledges the use of functionality within the Netcool/OMNIbus Probe for Ping that was developed by Stanford University.

Netcool/SM Operations, Netcool/SM Configuration, and Netcool/OMNIbus probes and monitors include software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/. Copyright (c) 1998-2005 The OpenSSL Project. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)"

4. The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org.

5. Products derived from this software may not be called "OpenSSL" nor may "OpenSSL" appear in their names without prior written permission of the OpenSSL Project.

6. Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/)"

THIS SOFTWARE IS PROVIDED BY THE OpenSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

Original SSLeay License Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com). All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscapes SSL. This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com). Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. All advertising materials mentioning features or use of this software must display the following acknowledgement: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)".

4. If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG ``AS IS'' AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION)

HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publicly available version or derivative of this code cannot be changed, i.e. this code cannot simply be copied and put under another distribution licence [including the GNU Public Licence.]

Micromuse acknowledges the use of software developed by ObjectPlanet. ©2003 ObjectPlanet, Inc., Ovre Slottsgate, 0157 Oslo, Norway.

Micromuse acknowledges the use of Expat in Netcool/ASM. Copyright 1998, 1999, 2000 Thai Open Source Software Center Ltd. and Clark Cooper. Copyright 2001, 2002 Expat maintainers. THE EXPAT SOFTWARE IS PROVIDED HEREUNDER "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS OF THE EXPAT SOFTWARE BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE EXPAT SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE. Expat explicitly grants its permission to any person obtaining a copy of any Expat software and associated documentation files (the "Expat Software") to deal in the Expat Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Expat Software. Expat's permission is subject to the following conditions: The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Expat Software. Except as set forth hereunder, all software provided by Micromuse hereunder is subject to the applicable license agreement.

Micromuse acknowledges that Netcool Security Manager includes Hypersonic SQL. Copyright (c) 2001-2002, The HSQL Development Group. All rights reserved.

JABBER® is a registered trademark and its use is granted under a sublicense from the Jabber Software Foundation.

Micromuse acknowledges the use of MySQL in Netcool/Precision for IP Networks and in Netcool/Precision for Transmission Networks. Copyright © 1995, 1996 TcX AB & Monty Program KB & Detron.

Micromuse acknowledges the use of Cryptix in Netcool/Precision IP. Copyright (c) 1995-2004 The Cryptix Foundation Limited. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE CRYPTIX FOUNDATION LIMITED AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE CRYPTIX FOUNDATION LIMITED OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Micromuse acknowledges the use of PCRE in Netcool/Precision. Copyright ©1997-2005 University of Cambridge. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. Neither the name of the University of Cambridge nor the name of Google Inc. nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE

POSSIBILITY OF SUCH DAMAGE.

Micromuse acknowledges the use of Net-SNMP in Netcool/ISM and Netcool/OMNIbus probes & monitors.

Part 1: CMU/UCD copyright notice: (BSD like) Copyright 1989, 1991, 1992 by Carnegie Mellon University Derivative Work - 1996, 1998-2000. Copyright 1996, 1998-2000 The Regents of the University of California. All Rights Reserved. Permission to use, copy, modify and distribute this software and its documentation for any purpose and without fee is hereby granted, provided that the above copyright notice appears in all copies and that both that copyright notice and this permission notice appear in supporting documentation, and that the name of CMU and The Regents of the University of California not be used in advertising or publicity pertaining to distribution of the software without specific written permission.

CMU AND THE REGENTS OF THE UNIVERSITY OF CALIFORNIA DISCLAIM ALL WARRANTIES WITH REGARD TO THIS SOFTWARE, INCLUDING ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS. IN NO EVENT SHALL CMU OR THE REGENTS OF THE UNIVERSITY OF CALIFORNIA BE LIABLE FOR ANY SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES OR ANY DAMAGES WHATSOEVER RESULTING FROM THE LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR OTHER TORTIOUS ACTION, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Part 2: Networks Associates Technology, Inc. copyright notice (BSD) Copyright (c) 2001-2003, Networks Associates Technology, Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Networks Associates Technology, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Part 3: Cambridge Broadband Ltd. copyright notice (BSD) Portions of this code are copyright (c) 2001-2003, Cambridge Broadband Ltd. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- The name of Cambridge Broadband Ltd. may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDER “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDER BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Part 4: Sun Microsystems, Inc. copyright notice (BSD) Copyright © 2003 Sun Microsystems, Inc., 4150 Network Circle, Santa Clara, California 95054, USA. All rights reserved. Use is subject to license terms below. This distribution may include materials developed by third parties. Sun, Sun Microsystems, the Sun logo and Solaris are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of the Sun Microsystems, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF DVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Part 5: Sparta, Inc. copyright notice (BSD) Copyright (c) 2003-2004, Sparta, Inc. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Sparta, Inc. nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

Part 6: Cisco/BUPTNIC copyright notice (BSD) Copyright (c) 2004, Cisco, Inc. and Information Network, Center of Beijing University of Posts and Telecommunications. All rights reserved. Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

- Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

- Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

- Neither the name of Cisco, Inc., Beijing University of Posts and Telecommunications, nor the names of their contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS” AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT HOLDERS OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF

SUCH DAMAGE.

Micromuse Inc., 650 Townsend Street, San Francisco, USA CA 94103

www.micromuse.com

Document Version Number: 1.0

Contents

Preface

This guide describes how to install, administer, and use Netcool for Wireless User Quality. The following chapters and appendixes describe each functional area, and task-oriented examples are provided to assist users and administrators in configuring and using the application.

This preface contains the following sections:

Audience

This guide is inteded for administrators within wireless service provider organizations. It provides detailed information about installing and configuring Netcool for Wireless User Quality on end user handsets. In addition, it is designed to be used as a reference guide to assist you in designing and configuring your environment.

Netcool for Wireless User Quality works in conjunction with Netcool®/OMNIbus™ and Netcool®/Webtop™. It is assumed that you understand how these products work. For more information on Netcool/OMNIbus and Netcool/Webtop, refer to the publications described in *Associated Publications* on page 4.

About this Guide

This book is organized as follows:

- Chapter 1: *Introduction* on page 11 provides an overview of the Netcool for Wireless User Quality Monitoring unit components.
- Chapter 2: *Installation and Start-up* on page 31 describes how to install the Netcool for Wireless User Quality Monitoring unit components, including information on starting up and removing them.
- Chapter 3: *Configuration* on page 41 provides guidelines on how to create and set values in the XML configuration file which determines the settings for the Monitoring unit components.
- Chapter 4: *Monitoring Event Information* on page 89 describes the measurements provided by the Netcool for Wireless User Quality Monitoring unit, including details of information available with the different monitor types.
- Appendix A: *XML File Example* on page 123 provides a full example of a valid XML configuration file for the Netcool for Wireless User Quality Monitoring unit.

Associated Publications

This section describes the documentation associated with Netcool for Wireless User Quality. To efficiently administer Netcool for Wireless User Quality, you must also understand Netcool/OMNIbus and Netcool/Webtop technology.

Netcool®/OMNIbus™ Installation and Deployment Guide

This book is intended for Netcool administrators who need to install and deploy Netcool/OMNIbus. It includes installation, upgrade, and licensing procedures. In addition, it contains information about configuring security and component communications. It also includes examples of Netcool/OMNIbus architectures and how to implement them.

Netcool®/OMNIbus™ User Guide

This book is intended for anyone who needs to use Netcool/OMNIbus desktop tools on UNIX or Windows platforms. It provides an overview of Netcool/OMNIbus components, as well as a description of the operator tasks related to event management using the desktop tools.

Netcool®/OMNIbus™ Administration Guide

This book is intended for system administrators who need to manage Netcool/OMNIbus. It describes how to perform administrative tasks using the Netcool/OMNIbus Administrator GUI, command line tools, and process control. It also contains descriptions and examples of ObjectServer SQL syntax and automations.

Netcool®/OMNIbus™ Probe and Gateway Guide

This guide contains introductory and reference information about probes and gateways, including probe rules file syntax and gateway commands. For more information about specific probes and gateways, refer to the documentation available for each probe and gateway on the support site.

Online Help

Netcool GUIs contain either context-sensitive online help with index and search capabilities, or online documentation, HTML versions of the associated guides.

Netcool®/Webtop Administration Guide

This guide is intended for administrators who need to manage the Netcool/Webtop Graphical User Interface. It describes how to install, administer and use Netcool/Webtop.

Netcool® GUI Foundation Administration Guide

This book describes how to administer the Netcool GUI Foundation, the central server application that runs GUIs from different Netcool products. It describes how to configure the Netcool GUI Foundation server, manage users, create and provision pages, and administer security permissions.

Netcool for Wireless User Quality® Core Unit Administration Guide

This guide is intended for administrators who need to install, configure, and manage the Core unit components of the Netcool for Wireless User Quality solution.

Typographical Notation

Table 1 on page 6 shows the typographical notation and conventions used to describe commands, SQL syntax, and graphical user interface (GUI) features. This notation is used throughout this book and other Netcool® publications.

Table 1: Typographical Notation and Conventions (1 of 2)

Example	Description
`Monospace`	The following are described in a monospace font: • Commands and command line options • Screen representations • Source code • Object names • Program names • SQL syntax elements • File, path, and directory names Italicized monospace text indicates a variable that the user must populate. For example, `-password password`.
Bold	The following application characteristics are described in a bold font style: • Buttons **Note:** Text in the pop-up tooltips is used to name buttons with icons. These button names are described in plain text. • Frames • Text fields • Menu entries A bold arrow symbol indicates a menu entry selection. For example, **File→Save**.
Italic	The following are described in an italic font style: • An application window name; for example, the *Login* window • Information that the user must enter • The introduction of a new term or definition • Emphasized text • References to external documents

[1]	Code or command examples are occasionally prefixed with a line number in square brackets. For example:
	`[1] First command...`
	`[2] Second command...`
	`[3] Third command...`

Table 1: Typographical Notation and Conventions (2 of 2)

Example	Description
`{ a | b }`	In SQL syntax notation, curly brackets enclose two or more required alternative choices, separated by vertical bars.
`[ ]`	In SQL syntax notation, square brackets indicate an optional element or clause. Multiple elements or clauses are separated by vertical bars.
|	In SQL syntax notation, vertical bars separate two or more alternative syntax elements.
`...`	In SQL syntax notation, ellipses indicate that the preceding element can be repeated. The repetition is unlimited unless otherwise indicated.
`,...`	In SQL syntax notation, ellipses preceded by a comma indicate that the preceding element can be repeated, with each repeated element separated from the last by a comma. The repetition is unlimited unless otherwise indicated.
a	In SQL syntax notation, an underlined element indicates a default option.
`( )`	In SQL syntax notation, parentheses appearing within the statement syntax are part of the syntax and should be typed as shown unless otherwise indicated.

Many Netcool commands have one or more command line options that can be specified following a hyphen (`-`).

Command line options can be `string`, `integer`, or `BOOLEAN` types:

- A `string` can contain alphanumeric characters. If the string has spaces in it, enclose it in quotation (`"`) marks.
- An `integer` must contain a positive whole number or zero (`0`).
- A `BOOLEAN` must be set to `TRUE` or `FALSE`.

SQL keywords are not case-sensitive, and may appear in uppercase, lowercase, or mixed case. Names of ObjectServer objects and identifiers are case-sensitive.

Note, Tip, and Warning Information

The following types of information boxes are used in the documentation:

Note: Note is used for extra information about the feature or operation that is being described. Essentially, this is for extra data that is important but not vital to the user.

Tip: Tip is used for additional information that might be useful for the user. For example, when describing an installation process, there might be a shortcut that could be used instead of following the standard installation instructions.

Warning: Warning is used for highlighting vital instructions, cautions, or critical information. Pay close attention to warnings, as they contain information that is vital to the successful use of our products.

Syntax and Example Subheadings

The following types of constrained subheading are used in the documentation:

Syntax

Syntax subheadings contain examples of ObjectServer SQL syntax commands and their usage. For example:

```
CREATE DATABASE database_name;
```

Example

Example subheadings describe typical or generic scenarios, or samples of code. For example:

```
[1]   <body>
[2]      <img src="ChartView?template=barchart&format=PNG
[3]       &request=image&chart=quote&width=800&height=400" border="0" height="400"
[4]       width="800" alt="Events by Severity"
[5]      >
[6]   </body>
```

Operating System Considerations

All command line formats and examples are for the standard UNIX shell. UNIX is case-sensitive. You must type commands in the case shown in the book.

Unless otherwise specified, command files are located in the `$OMNIHOME/bin` directory, where `$OMNIHOME` is the UNIX environment variable that contains the path to the Netcool for Wireless User Quality home directory.

On Microsoft Windows platforms, replace `$OMNIHOME` with `%OMNIHOME%` and the forward slash (`/`) with a backward slash (`\`).

Chapter 1: Introduction

This chapter provides an overview of the Netcool for Wireless User Quality solution's Monitoring unit. For an overview of the entire Netcool for Wireless User Quality architecture, refer to the *Netcool for Wireless User Quality Core Unit Administration Guide*.

This chapter contains the following sections:

1.1 Overview

Netcool for Wireless User Quality is a solution that enables wireless service providers to monitor and analyze the service quality delivered to customers. The Netcool for Wireless User Quality Monitoring unit comprises the data collection and forwarding components of the overall solution. The Monitoring unit components run on mobile phone handsets, passively collecting information on various aspects of the end-user's experience of the service. The information is then forwarded to the Netcool for Wireless User Quality Core unit for processing and analysis.

The Netcool for Wireless User Quality Monitoring unit consists of the following components:

- Control Function
- Voice Monitor
- Data Monitor
- Video Monitor
- Active Client

The Control Function manages the overall operation of the monitors. It is responsible for configuring, starting up, and shutting down the monitors. It also gathers events from the monitors and transmits them to the Netcool for Wireless User Quality Core unit. The monitors measure the performance of a service from the user's perspective by collecting data about an activity in the network directly on the handset (for example, a voice call). Using the performance data, the monitors generate events which are messages containing the measurement results.

The Monitoring unit operates in the background without any need for the handset user to interact with it. In fact, for all monitors except the Active Client, the handset user does not notice the monitoring operation.

All monitors can operate at the same time. A handset can have all monitor types installed and running simultaneously, producing results without affecting each other's operation.

The Monitoring unit supports the Symbian OS platform. For information on which version(s) of the Symbian OS are supported by each monitor, see *Prerequisites* on page 32.

The following sections provide an overview of the functions and features provided by the Netcool for Wireless User Quality Monitoring unit components. For an overview of the entire solution's architecture, including the Core unit, see the *Netcool for Wireless User Quality Core Unit Administration Guide*.

1.2 About the Control Function

The Control Function is the component of the Monitoring unit that manages the service monitors and the reporting of their events to the Core unit. The Control Function runs silently on each handset and its operation is not visible to users. Depending on the network, the Control Function uses technologies such as General Packet Radio Service (GPRS) or Universal Mobile Telecommunications System (UMTS) when transmitting events.

Before sending the events, the Control Function applies compression and AES encryption to them. The Netcool for Wireless User Quality Core unit's Event Collector is able to verify the origin of the events, and decompress and decode them.

The Control Function installation consists of a single file that you install onto your handset. For further information about installation, see *Installation and Start-up* on page 31.

The following sections describe the responsibilities the Control Function fulfills within the monitoring operation.

Configuration Handling

The Monitoring unit is configured using an XML configuration file. The file stores the configuration of the entire Monitoring unit, including settings for the Control Function itself and each of the monitor types. You set values in the XML file and then download the file to the handset using any file transfer mechanism (for example, an HTTP pull operation or a handset provisioning system). For downloading through an HTTP pull operation, which requires user interaction on the handset, the Control Function has a separate engineering GUI. This GUI has additional features:

- It provides the option to edit the data connection settings for the HTTP pull operation and a command to launch a fetch of the configuration file. For more information about setting the connection and fetching the configuration file, see *HTTP Pull* on page 82.
- It presents information on the status of the Monitoring unit components, including a list of monitors running, and a notification message if the configuration file is missing or invalid.
- It provides a `kill` command for stopping the Monitoring unit's main process, as described in *Uninstalling the Netcool for Wireless User Quality Monitoring Unit* on page 38.
- It provides a menu item called **Legal** for checking the legal message presented to the handset end user if an owner acceptance screen has been set up. The service provider can create a legal notice asking the end user to consent to the operation of performance measuring monitors on their handset before the monitor software becomes active. For more information on this option, see *Creating an Owner Acceptance Screen* on page 85.

You can specify whether the Control Function automatically checks a specified location for updated configuration files. If a newer file is available, it is downloaded using HTTP, and the Control Function overwrites the existing file. The new settings then take effect. If no new file is available, the handset continues to use the existing configuration.

If automatic configuration updates are enabled, you can specify an approximate interval, in seconds, between checks. To avoid many handsets contacting the server simultaneously, a random time is added to the interval you specify, so you cannot specify the exact timing of updates. However, you can specify a maximum value for the random time that is added to your chosen interval. For further information, see *The <configfetcher> Section (optional)* on page 43.

Note: As a service provider, you have the option of including your own default configuration file on handsets. Your file can contain either full handset configuration or only the information that allows the handset to download a current configuration file. For example, if your file contains only a `<configfetcher>` section, then no monitoring takes place, but you can control when the handset downloads a new configuration file. You can then make updated configuration files available for download at your convenience.

A valid configuration file must be present for the Monitoring unit to function. If the Control Function cannot read this file when the handset is switched on, it cannot read definitions for what to measure, and therefore cannot launch the monitoring operations. Also, the Control Function uses configuration file settings to detect how recently the file was downloaded. For these reasons, the configuration file must always have the same name and be stored in the same location. This storage is persistent, and restarting the handset does not remove the configuration settings.

All monitors check the validity of configuration parameters before launching the monitoring operation. For example, IP address formats are checked, numbers are verified as being numeric, and strings are checked for being within a particular range if there is a finite range. If a monitor detects invalid parameters, it does not start and a message is displayed on the engineering GUI about the problem. The engineering GUI retains messages, so if the GUI is not being accessed when a message is generated, it can be viewed at a later time.

In addition, the correct SIM card must be inserted in the handset for monitoring to be enabled. The Control Function compares the Mobile Country Code (MCC) and Mobile Network Code (MNC) stored on the SIM against those stored in the configuration file. If a match is not found, then the handset cannot connect to a valid network and monitoring is not enabled. If a valid network is found, and roaming is enabled, then the handset starts correctly. For further information, see *The <commprefselector> Section* on page 73.

Upon detection of the configuration file, the Control Function reads it, instructs the monitors defined in the file to start up, and provides the monitors with their settings from the file.

For details on how to configure and download the XML file, including the use of the Control Function's engineering GUI, see *Configuration* on page 41.

Buffering and Load Control

Monitors forward their results to the Control Function which is the single component that manages them. The Control Function queues the events if it is temporarily unable to transmit them to the Core unit (for example, due to a temporary loss of signal). The events are sent to the Core unit's Event Collector in packets at set intervals, and the packets are made up of events taken from the queue, as described in *Reporting* on page 16. You can set the reporting interval in seconds, with the minimum allowed interval being five seconds.

Note: The number of events placed in a packet, or in other words, a *report*, is determined by the size of the events. Although the events are compressed, both UDP and TCP packets have a size limit of 16,384 Bytes. This means that the number of events transmitted per packet depends heavily on the size of the events in the queue.

As events can arrive from monitors faster than the reporting interval, the queue can grow large, resulting in excessive use of handset memory. To avoid this, you can configure the maximum number of events stored in the queue, as described in *The <queue> Section* on page 69. When the maximum number of events in the queue is reached, the oldest event is discarded. This helps to maintain a real-time view of the service performance. The Control Function supports multiple reports, so when a report is full, another one can be created if there are more events in the queue. If there are several reports waiting to be sent, the oldest ones are sent first to ensure chronological order.

To reduce the size of events, both the Control Function and the Event Collector have access to a common default dictionary which has been designed to contain the most common repetitions. When the Control Function compresses events in a report, it uses dictionary definitions to encode certain strings with references. The Event Collector has access to the same dictionary and can therefore decode the references.

Warning: Do not replace the default dictionary with a custom dictionary. If you do, the Event Collector can longer read the references to the strings.

There is an option to set filters on events in order to further limit the network load and to allow you to focus on event information that is relevant. Filters determine which events to place in the queue as described in *Filtering* on page 16.

The buffering capability of the Control Function also helps in controlling the burst of events transmitted to the Core unit's Event Collector and in cases when a temporary problem occurs, such as loss of connectivity.

For details on how to set buffering and load control values, see *The <queue> Section* on page 69 and the *The <maxageseconds> Section (optional)* on page 71.

Reporting

The Control Function is responsible for sending groups of events (reports) to the Event Collector. When an event is generated, the Control Function initiates an Internet connection using a predefined Internet Access Point (IAP). Depending on the mobile network and the settings defined in the configuration file, you can control how the Control Function connects to the Internet, as described in *The <iap> Section* on page 71.

Once the handset is connected to the Internet, a connection to the Event Collector is established. This connection uses either the UDP or TCP transport protocol, depending on settings in the configuration file. For information on configuring the reporting to the Event Collector, see *The <reporter> Section* on page 75. Events are packaged into reports up to the maximum size of 16,384 Bytes, and sent to the Event Collector.

The Control Function manages connections by reporting events as efficiently as possible when a connection exists. This reduces the number of times connections need to be made and conserves handset battery power. Also, by reporting events quickly, correlation with location data is possible.

Note: If an existing data connection is open, the Control Function can only send events if the handset is a GPRS class A device or a 3G handset, capable of multiple simultaneous data transfer sessions. If the handset is not a class A or 3G device, then the events are queued by the Control Function, and they are only transmitted when the current data session terminates.

If new events are made available for reporting while the connection is still open, then multiple reports can be sent using the existing connection.

To preserve battery power, you can configure a minimum interval between the end of one connection and the start of the next. Similarly, in the event of a failed connection, the Control Function is designed to incorporate an interval before attempting another connection. If there are repeated failed attempts to connect, the rate at which connection attempts are made is reduced, so that there is a longer interval between successive retries. You can configure a maximum interval duration, as described in *The <iptransmitter> Section* on page 76.

Filtering

The Control Function can filter events by comparing actual event tokens with the settings defined in the configuration file. Based on the comparison, the Control Function can reject an event. For example, you may choose to only report events that indicate a service problem, and discard those that do not.

You can use various criteria to filter events. For example, you can specify thresholds for specific Key Performance Indicator (KPI) values for each monitor type, such as a Mean Opinion Score (MOS) for the Voice Monitor. If the MOS goes above or below the specified value, then the event is rejected.

Since older events are less useful in providing a real-time representation of the network, you can specify the maximum age of the events in the XML configuration file as described in *The <maxageseconds> Section (optional)* on page 71. Events older than the time specified are not transmitted. Additionally, if the queue is full, oldest events are discarded first by default to make way for new events, as described in *The <queue> Section* on page 69.

To get a true representation of usage, you can disable filtering completely, allowing all events to be reported.

Note: You can only disable filtering for all monitors. You cannot, for example, enable filtering for one monitor and disable it for another.

You can set filtering options in a separate section in the configuration file. For more information on setting filters, see *The <filter> Section (optional)* on page 64.

Handset Information

In addition to the information provided by the monitors, the Control Function can determine and add key details of the handset device and the connection to the event data, including the following:

- International Mobile Equipment Identity (IMEI)
- International Mobile Subscriber Identity (IMSI)
- Current date and time
- Handset model and vendor
- Handset's operating system type and version
- Global Positioning System (GPS) location data (if the Control Function connects to a separate GPS device through Bluetooth)

Note: The Control Function can connect to any GPS device based on the settings described in *The <positionfinder> Section (optional)* on page 70. The Control Function maintains a continuous connection with the GPS device and places the location data attained from the device into the events to be transmitted.

For further details of the data added to the events by the Control Function, see *Control Function Measurements* on page 90.

Roaming Support

The Control Function is able to send events to the Netcool for Wireless User Quality Core unit even when the handset is roaming. This helps the network operator to assess the service quality level provided by the networks of other operators with which they have roaming agreements.

The Control Function can be set to recognize when the handset is roaming and whether it needs to report for the specific country. This requires a list of operator long names in order to determine which Access Point Name (APN) should be dialed in which country. The Control Function can store operator long names and the APNs associated to them as reference.

For details on how to configure roaming, see *The <waitfornet> Section (optional)* on page 46 and *The <iap> Section* on page 71.

1.3 About the Voice Monitor

The Voice Monitor measures the quality of voice calls. It passively collects information about the received voice quality on the handset. The Voice Monitor uses the Psytechnics NiQA algorithm to assess voice quality. The NiQA algorithm monitors the voice waveform and produces voice quality measurement scores based on the International Telecommunications Union (ITU) Mean Opinion Score (MOS) scale. The scale represents the customers' perceptions of service quality and together with filtering, the scores can be used to trigger network alarm thresholds and give continuous feedback on the quality of the network. The Voice Monitor generates handset location information such as Location Area Code (LAC) and Cell ID with all events, so you can use event data in a Geographic Information System (GIS).

The Voice Monitor is designed to report any errors to the Netcool for Wireless User Quality Core unit. For example, if a call fails to connect, the Voice Monitor sends an event to the Control Function, specifying the reason for failure. Similarly, if the NiQA algorithm fails, an event is generated to indicate this error. Using such error reporting, you are kept informed of any problems and can take appropriate action to rectify Voice Monitor operation.

Note: For further information on the Psytechnics NiQA algorithm, visit http://www.psytechnics.com.

If the Voice Monitor is being used when a call is put on hold to take another call, monitoring stops when the call is put on hold. Neither of the two active calls are monitored further. Monitoring recommences at the start of the next call after both active calls have ended. This is shown in Figure 1.

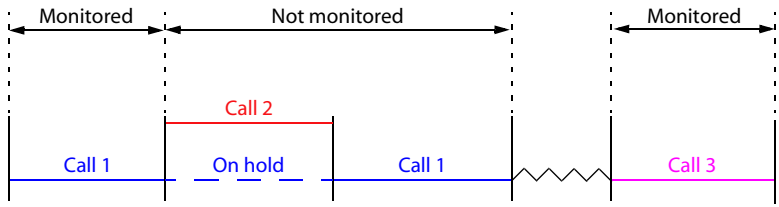

Figure 1: Voice Monitor Operation with Held Calls

The Voice Monitor also works when a handset's mute function is activated.

The Voice Monitor has three analysis modes: *Short*, *Full* and *Continuous*, as described in the following sections.

Short Analysis Mode

The *Short* analysis mode provides a fast, real-time view of the call quality. The short length of the analysis ensures that an event is sent to the Event Collector in real-time, providing rapid receipt of voice quality assessment. The short length of the sampling period minimizes the impact on the performance of the handset in terms of battery usage.

Note: In Short analysis mode, the results can only be sent during the active call if the handset is a GPRS class A device or a 3G handset, capable of simultaneous voice and data transfer sessions. If the handset is not a class A or 3G device, then the events are queued by the Control Function, and they are only transmitted after the call terminates.

In addition to information common to all monitors, as described in *Measurements Common to all Monitors* on page 92, the Voice Monitor reports the following information in Short analysis mode:

- Assessment mode (in this case, this value is Short)
- MOS
- Noise level
- Signal to noise ratio
- Level compensated MOS
- Speech activity
- Speech level
- Number of frames before speech starts
- Tone diagnostic code

For details of the information provided by the Voice Monitor's Short analysis mode, see *Voice Monitor Measurements* on page 94.

For information on defining the Voice Monitor's Short analysis mode settings in the XML configuration file, see *Example of a Short Mode Configuration* on page 49.

Full Analysis Mode

The *Full* analysis mode provides a more detailed quality analysis for the full duration of the call. The downlink voice waveform is sampled at defined regular intervals and then averaged for the duration of the entire call. The results are sent to the Netcool for Wireless User Quality Core unit upon the completion of the call. The Full analysis mode delivers comprehensive data collection spanning the whole length of the call, enabling a full quality assessment from the start of the call to its termination. This does mean, however, that as the length of the call cannot be known, the quality assessment cannot be reported in near real-time. As a result, the Full analysis mode is useful for engineering purposes and also for performing a detailed examination of the service quality provided.

In addition to information common to all monitors, as described in *Measurements Common to all Monitors* on page 92, the Voice Monitor reports the following information in Full analysis mode:

- Assessment mode (in this case, this value is Full)
- Average, minimum, and maximum MOS
- Average, minimum, and maximum noise level
- Average, minimum, and maximum signal to noise ratio
- Call duration
- Level compensated MOS
- Speech activity
- Speech level
- Number of frames before speech
- Tone diagnostic code

For details of the information provided by the Voice Monitor's Full analysis mode, see *Voice Monitor Measurements* on page 94.

For information on defining the Voice Monitor's Full analysis mode settings in the XML configuration file, see *Example of a Full Mode Configuration* on page 49.

Continuous Analysis Mode

The *Continuous* analysis mode is similar to the Short analysis mode, but instead of a single result being transmitted to the Control Function, results are transmitted periodically throughout the call. The downlink voice waveform is sampled at defined regular intervals and the results are sent to the Netcool for Wireless User Quality Core unit to provide a rolling analysis of voice quality.

Note: As in Short analysis mode, results from Continuous analysis mode can only be sent during the active call if the handset is a GPRS class A device or a 3G handset, capable of simultaneous voice and data transfer sessions. If the handset is not a class A or 3G device, then the events are queued by the Control Function, and they are only transmitted after the call terminates.

The Voice Monitor reports the same information in Continuous analysis mode as that in Short analysis mode:

- Assessment mode (in this case, this value is Continuous)
- MOS

- Noise Level
- Signal to noise ratio
- Level compensated MOS
- Speech activity
- Speech level
- Number of frames before speech starts
- Tone diagnostic code

For details of the information provided by the Voice Monitor's Continuous analysis mode, see *Voice Monitor Measurements* on page 94.

For information on defining the Voice Monitor's settings in the XML configuration file, see *Example of a Continuous Mode Configuration* on page 50.

1.4 About the Data Monitor

The Data Monitor is designed to measure the quality of service that users experience when using a web browser on a mobile handset. You can monitor HTTP content, such as specific web pages, file types, or file names. For example, you can collect information about `.mp3` file downloads. You can also view status information about web pages and web page components such as graphics and other content. For example, you can see if a page has loaded properly or not. If it has not loaded, you can find out why by examining the status codes contained in the report.

The Data Monitor collects information passively by listening for HTTP data communication packets between the handset browser and the server providing the content, and analyzing their payload. Normally, a single event is generated for each web page.

You can also monitor the download of HTML pages from a specific IP address, or from a range of IP addresses. Alternatively, you can filter out content from a particular IP address. See *Data Monitor Settings* on page 52 for further information about monitoring particular IP addresses.

Note: If a page contains links that are generated automatically using JavaScript, more than one event is generated for the page. For example, some pages may contain hyperlinks that are activated if a particular option is selected on a form. See *Automated Links and Dynamic Content* on page 104 for further information about automated links.

After analysis, the Data Monitor sends events to the Control Function for reporting to the Netcool for Wireless User Quality Core unit. As with the Voice Monitor, the Data Monitor generates handset location information, including LAC and Cell ID, with all events.

In addition to information common to all monitors, as described in *Measurements Common to all Monitors* on page 92, the Data Monitor reports the following information:

- URL of the HTTP data communication
- Data rate
- Number of received and sent bytes
- Number of components on page
- Number of components downloaded successfully
- Total time of HTTP transaction
- Protocol type and version
- Handset IP address and port
- Content server IP address and port

- Timeout value for data communication
- HTTP parsing level and status

For full details of the information provided by the Data Monitor, see *Data Monitor Measurements* on page 104.

Standards and Technologies

The Data Monitor supports the following standards and technologies:

Note: On Nokia handsets, the Data Monitor supports only the default Nokia browser.

- Detection of TCP/IP protocol packets.
- Traditional HTTP/1.0 transaction monitoring.
- Latest HTTP/1.1 transaction monitoring. This includes connection Keep-Alive Requests and chunked encoding content detection and monitoring. Chunked encoding facilitates the retrieval of content of unknown size, by providing auto-generated data in chunks. This is useful when the size of the data is unknown, for example, on dynamically generated web pages.
- Request pipelining, allowing multiple HTTP requests to be sent at once, without waiting for an answer. Request pipelining is supported on HTTP/1.1.
- HTTP command support. The `GET` and `POST` queries are monitored, while `HEAD` queries are not as they do not provide any context to monitor.
- Gzip compression for downloaded data.
- WAP 2.x transaction monitoring, as it uses HTTP over TCP/IP.
- I-mode transaction monitoring on TCP/IP using HTTP/1.0 or HTTP/1.1.

Note: The WAP 1.x protocol is not supported as it does not use HTTP over TCP/IP.

- The monitoring of SSL connections under HTTPS are not supported using HTTP monitoring, as the content is encrypted. However, you can monitor SSL connections using the Packet Statistics feature.

The Data Monitor also has two additional features: *Packet Statistics* and *Interface Monitoring*, described in the following sections. You can enable or disable these features as required.

You can configure all aspects of the Data Monitor operation, including *Packet Statistics* and *Interface Monitoring*, using the XML configuration file. For information on defining the Data Monitor settings in this XML configuration file, see *Data Monitor Settings* on page 52.

Packet Statistics Monitoring

This feature provides statistical information about packet-based services such as streaming video content or FTP downloads. You can use Packet Statistics monitoring to provide information about UDP packets, TCP packets, or both. You enable or disable Packet Statistics monitoring in the XML configuration file, as described in *Example of Using the Packet Statistics Feature* on page 54.

For example, you can obtain continuous analysis of packets received from a live video stream at a specific IP address. Depending on your requirements, you can configure the Data Monitor to report information about the stream in real time, either at specific intervals for a set duration, or only at the end of a given period, or both.

Enabling Packet Statistics monitoring allows you to receive the following information:

- Statistical information about variations in delivery delay (jitter)
- Number of packets out of sequence
- Number of duplicate packets
- Total number of packets through connection
- Packet size information
- Protocol used
- Handset's IP address and port
- Content server IP address and port

Interface Monitoring

This feature provides information about failed connections. You can enable or disable Interface Monitoring in the XML configuration file, as described in *Example of Using the Interface Monitoring Feature* on page 55.

If a handset cannot connect to the network, the Data Monitor sends an event to the Control Function, provided Interface Monitoring is enabled.

Events are also sent to the Control Function for all successful connections. These events contain the name assigned to the connection and the time taken to establish the connection.

Enabling Interface Monitoring generates events containing the following information:

- Interface name of a failed connection
- The error that caused the failure

- Successful connection name
- Time taken to establish the connection

1.5 About the Video Monitor

The Video Monitor uses Psytechnics Video IP (PVI) Monitor to analyze the quality of video sessions on handsets. The PVI software provides video quality measurement scores and diagnostic information on both incoming and outgoing traffic. These quality scores are analogous to the voice quality scores produced by the Voice Monitor.

As with the other monitors, the Control Function can only start the Video Monitor if a valid configuration is defined in the XML configuration file. You can control the operation of the Video Monitor using this file. If there is any problem with the configuration, the monitor is not started. If the Video Monitor configuration is valid, the PVI client is started.

As with the Data Monitor, the Video Monitor can monitor content from specific IP addresses, based on settings from the configuration file. See *Video Monitor Settings* on page 59 for more information on IP filtering. The PVI software notifies the Video Monitor of any new incoming video streams, and if the stream is to be monitored, it is placed in a queue for sending to the PVI software for analysis.

After analysis, the PVI software sends results to the Video Monitor so that events can be sent to the Control Function.

The Video Monitor returns two types of events: stream events and substream events. A substream is part of a stream, so events generated while a video stream is still running are based on a substream. Such events contain information about the portion of the video stream monitored just before the event is generated, and are sent to the Control Function immediately. Substream events are useful in providing periodic results. You can disable the generation of substream results by setting the sample period to zero (0) in the configuration file.

Note: Most video monitor measurements can be used for both streams and substreams, as described in *Video Monitor Measurements* on page 114.

When a stream ends, the PVI server notifies the Video Monitor and sends final stream results. The Video Monitor then generates an event containing the full stream results and sends it to the Control Function.

For information on defining the Video Monitor settings in the XML configuration file, see *Video Monitor Settings* on page 59.

In addition to information common to all monitors, as described in *Measurements Common to all Monitors* on page 92, some of the information reported by the Video Monitor is as follows:

- Type of stream
- Duration of stream
- Transport protocol

- Packet delivery delay variations (jitter)
- Number of packets expected and lost
- Average number of consecutive packets lost
- Length of continuous streams and gaps in video streams
- MOS of video stream
- Number of diagnostic packets
- Quality degradation analysis measurements
- Congestion analysis measurements
- Frame information
- Delay information
- Network address information

For full information about these, and other, video monitor measurements, see *Video Monitor Measurements* on page 114.

1.6 About the Active Client

The Active Client provides controlled monitoring of network service quality by scheduling periodic calls, and data and video sessions. The Active Client is used *in conjunction with* the Voice Monitor, the Data Monitor, and the Video Monitor.

As with the other monitors, the Active Client is started by the Control Function and has its own section in the configuration file, as described in *Active Client Settings* on page 60. However, the Active Client is different from the other monitors in that it is not a passive monitor, so it does not generate events. Instead, the Active Client is designed to simulate user operation of a handset by executing a sequence of actions that you can define in the configuration file. Based on the settings in the XML configuration file, the Active Client triggers the handset to perform calls or data sessions, which can then be measured by the other performance monitors. Thus, it is important to have the appropriate monitors installed and configured on the handset in order to make meaningful use of the Active Client.

The sequences can consist of any combination of voice, HTTP, WAP/I-Mode, Real-time Transport Protocol (RTP) or Real-time Streaming Protocol (RTSP) actions. To prevent sequences from overlapping, they are separated by a configurable inter-execution delay. Although sequences cannot overlap, actions within sequences can. For example, you can configure a sequence so that a voice call starts at the same time as a web download, but two voice calls cannot be launched at the same time. There is a limit of five concurrent actions, so if you configure more than five actions to start at the same time, only five will start. A 10 second delay is introduced before any more actions are started.

Note: When first initialized by the Control Function, the Active Client starts executing its sequence immediately. However, if the first action is a voice action, it is not monitored by the Voice Monitor. This is normal behavior, and is due to the fact that the Voice Monitor has not started at this point. This situation arises each time you reconfigure the monitors.

Action sequences always execute exactly as you specify in the configuration file, as described in *Active Client Settings* on page 60. No actions or sequences are dependent on any previous actions.

The Active Client has to be used together with the different monitors in the following ways:

- The Active Client can initiate calls and end calls. When making voice calls, the Active Client dials the number that you specify in the configuration file. If the Active Client tries to initiate a voice call while an existing call is open, the attempt is rejected. Also, if the remote party hangs up during a call, the configured sequence continues as normal, without interruption. During the call, the Voice Monitor can measure the quality of the call, as described in *About the Voice Monitor* on page 19.

Tip: To receive meaningful events from the Voice Monitor when initiating a voice call, set the configuration of the Active Client to call numbers that have recorded messages, such as voice mail.

- You can configure the Active Client to make the handset access web pages as part of its operation. In order to simulate user actions, the Active Client uses the `GET` request, an HTTP mechanism used to fetch web pages. The Data Monitor can then collect data about the data transmission as described in *About the Data Monitor* on page 23.
- To monitor video sessions, the Active Client connects to an RTP video stream and allows it to run for a configurable period of time. This forces the Video Monitor to monitor the stream and to generate events, as described in *About the Video Monitor* on page 27.

An optional splash screen, advising users that monitoring is in progress, can be displayed when the Active Client is operating. The screen displays the message `"Monitoring in progress"`, and includes an animation and a logo. You can enable or disable this screen, and control how it displays, using the settings in the configuration file, as described in *Active Client Settings* on page 60.

Chapter 2: Installation and Start-up

This chapter describes how to install and start up the Netcool for Wireless User Quality Monitoring unit components. It also provides information on the removal of the Monitoring unit from the handset.

This chapter contains the following sections:

2.1 Prerequisites

This section provides information on system prerequisites for the Netcool for Wireless User Quality Monitoring unit.

Supported Platforms

The Netcool for Wireless User Quality Monitoring unit supports the Symbian OS version 7.0s, 8.0, and 8.1 Series 60 platforms. The platform support for each monitor is as follows:

- Voice Monitor — Symbian OS version 7.0s, 8.0, and 8.1 Series 60
- Data Monitor — Symbian OS version 7.0s, 8.0, and 8.1 Series 60
- Video Monitor — Symbian OS version 8.0, and 8.1 Series 60
- Active Client — Symbian OS version 7.0s, 8.0, and 8.1 Series 60

For a complete list of supported handset types, please contact your account manager.

Note: There are known issues with the Monitoring unit affecting selected handsets with Symbian OS version 7.0s installed.

Handset Memory Requirement

The Monitoring unit components are installed to the handset memory. Ensure that there is a minimum of 1 MByte free space available in the handset memory.

Note: If the memory on the handset runs low, the handset may stop the Monitoring unit components to make memory space available for making calls.

Downloading the Installation Files

You can download the Monitoring unit components as a single package from the IBM support site. Download the file `wuq_2_0_monitors.zip`, and unpack its contents to your PC. The following files are included:

- `netcool.sis`
- `voicemonitor.sis`
- `datamonitor.sis`
- `videomonitor.sis`

- `activeclient.sis`
- `showgui.sis`
- `uqmconfig.pkg`

2.2 Installation

The Netcool for Wireless User Quality Monitoring unit uses the standard Symbian packaging and installation process. The Monitoring unit components are packaged into `.sis` files. A `.sis` file is a compressed installation package including all the necessary executables and other files needed to install the software on a Symbian device. It also provides information for the Symbian OS on how to install the application.

You install `.sis` files from your PC to your handset memory, using either Bluetooth or a wired connection, such as a USB cable. If you are using a wired connection, you may need to install driver software before using the connection. For further information, see your handset or driver software documentation.

You can also download and install `.sis` files onto handsets from a web location, after making the files available on your web server.

Since the Monitoring unit can be installed on different handsets, each having different features and menus, the exact installation procedure may vary. It is beyond the scope of this document to describe the exact procedure for all handsets. The procedure described is therefore for general guidance only, although the process should be similar for most handsets.

Note: The following sections often require you to perform tasks using the handset's menu items or computer software provided with the handset. If in doubt about how to use your handset or install software on it, please see the handset's user guide. Examples are provided for Nokia handset models 6630 and 6680 as both have the same menu structure.

Installing the Netcool for Wireless User Quality Monitoring Unit

To install the Netcool for Wireless User Quality Monitoring unit, follow these general steps:

1. Before starting the installation process, enable software installation on the handset. Also, turn online certificate checking off, as the Netcool for Wireless User Quality Monitoring unit is not Symbian certified.

   For example, do the following on the Nokia 6630 and 6680 handsets to be able to install the Monitoring unit:

   Select **Menu→Tools→Manager**. Select **Options→Settings** and check that the **Software installation** option is turned `on` and the **Online certificate check** is turned `off`.

2. Download the Netcool for Wireless User Quality Monitoring unit files, *in the order shown*, from your PC to the handset:

   - `netcool.sis` — Contains the Control Function software.
   - `voicemonitor.sis` — Contains the Voice Monitor software.

- `datamonitor.sis` — Contains the Data Monitor software.
- `videomonitor.sis` — Contains the Video Monitor software.
- `activeclient.sis` — Contains the Active Client software.
- `uqmconfig.sis` — Installs the XML configuration file to your handset. Include this file *only* if you have created the XML configuration file already and packaged it as described in *Packaging the Configuration File* on page 80.
- `showgui.sis` — Displays the engineering GUI password prompt. Include this file if you wish to access the engineering GUI as described in *Uninstalling the Netcool for Wireless User Quality Monitoring Unit* on page 38.

Note: The engineering GUI is used for several different functions as described in *Configuration Handling* on page 13.

Downloading to the handset can be performed through several connections depending on the capabilities of the handset and the software supplied for it.

For example, you can download the Monitoring unit `.sis` files onto the Nokia 6630 and 6680 handsets using Bluetooth, a connectivity cable, or by downloading files from a web server. See the handset-specific user guides for details.

Tip: If you have a Nokia handset, install the Nokia PC Suite software from the CD-ROM disk delivered with the handset. Once the Nokia PC Suite is installed, your computer recognizes the `.sis` extension. Then, use the Nokia PC Suite software to download the Netcool for Wireless User Quality Monitoring unit `.sis` files to the handset. Symbian OS handsets may have other management software, which enable the computer to handle `.sis` files.

3. Install the Monitoring unit files. Depending on whether you used handset management software from your computer (such as the Nokia PC Suite), or Bluetooth software, the installation starts immediately or after opening the downloaded `.sis` files:
   - If you used a handset management software installed on your computer, then the Symbian installer starts up on the handset right away, regardless of the connection.
   - If you downloaded the files via Bluetooth but not using the handset management software, then the `.sis` files arrive as messages in the handset's Inbox. In this case, open the `.sis` files to launch the installer.

Note: Install the Control Function software (`netcool.sis`) first. The other monitors may not install properly if the Control Function is not installed.

When the installation process begins, you are notified and asked to confirm that you want to install the Monitoring unit component. Select **Yes** to continue. You may see a message stating that the application might not be compatible with the handset and asking whether to quit the installation. Select **No** and continue the installation.

During the installation of the Monitoring unit to the handset memory, a progress bar is displayed. A notification message indicates when the installation is complete.

4. Reboot the handset to launch the installed Monitoring unit components.

   The Monitoring unit processes start up automatically after the handset starts.

   If you have used the packaging option for the configuration file (`uqmconfig.sis`) as described in step 2, the handset is now configured to use the settings in your custom configuration.

Warning: In this case, you must remove the `showgui.sis` installation if present as it may impact the end-user experience on the handset. Follow the steps in *Disabling the Engineering GUI* on page 36.

   If you have *not* used the packaging option for the configuration file, then you still need to download your XML configuration file to the handset. For information on defining settings in the XML file and downloading it to the handset, see *Configuration* on page 41.

Disabling the Engineering GUI

If you have used the engineering GUI to download your configuration file, you must disable the `showgui.sis` installation, as it may impact the end-user experience on the handset. Once you are satisfied that your handset is configured correctly, then disable `showgui.sis` as follows:

1. Locate the **ShowGui** component on the handset and delete it using the handset's menu system.

   For example, do the following on the Nokia 6630 and 6680 handsets to locate and remove the files:

   Select **Menu→Tools→Manager**. Use the arrow keys to select **ShowGui**. Select **Options→Remove**. You are asked to confirm your intention of removing the component. Select **Yes** to continue.

   During the removal process, a progress bar is displayed. A notification message is presented when the component has been removed.

2. Reboot the handset.

Re-installing a Monitor

You can re-install a monitor, or install a newer version of a monitor. The process involves stopping the Control Function, in order to prevent interference with the configuration or operation of other monitors.

Follow these steps to re-install a monitor:

1. Stop the Control Function.

Note: In order to stop the Control Function, the engineering GUI password prompt must be present on your handset. If you did not install this (by running `showgui.sis`) when you installed the monitors, then download and run `showgui.sis` now, following the steps in *Installing the Netcool for Wireless User Quality Monitoring Unit* on page 34. Ensure you reboot your handset.

2. Remove the existing monitor by following steps 1 to 4 of section *Removal* on page 38.

Warning: Do not reboot the handset at this point.

3. Follow the steps in *Installation* on page 34, as appropriate, to install the new monitor.

2.3 Removal

The Netcool for Wireless User Quality Monitoring unit components can be removed from the handsets.

Uninstalling the Netcool for Wireless User Quality Monitoring Unit

To uninstall the Netcool for Wireless User Quality Monitoring unit components, follow the steps below:

1. Enter the engineering GUI of the Monitoring unit on the handset.

Warning: In order to access the engineering GUI, the engineering GUI password prompt must be present on your handset. If you did not install this (by running `showgui.sis`) when you installed the monitors, then download and run `showgui.sis` now, following the steps in *Installing the Netcool for Wireless User Quality Monitoring Unit* on page 34. Ensure you reboot your handset.

For example, do the following on the Nokia 6630 and 6680 handsets to access the engineering GUI:

Use the arrow keys of the handset and enter the sequence of up, down, left, and right 3 times. If the sequence has been correctly entered, then the engineering GUI appears and asks for a password.

The engineering GUI's password is numeric, based on the current date and time as follows:

`DDMMYYYYTT`

where `DD` is day, `MM` is month, `YYYY` is year, `TT` is hour. For example, the password `2007200516` stands for a password provided when entering the engineering GUI on July 20th, 2005 at 16:19. For single digit morning hours, use a `0` for the first digit in the TT hour value. For example, if the time is 09:14 in the morning, the TT hour value should be specified as `09`.

Note: The `TT` hour value of the engineering GUI's password is based on local time.

Enter the date and time and press **OK**. If the entered details are correct, access is granted to the engineering GUI.

2. Select **Options→Kill**. This stops the Control Function's process.

3. Exit the engineering GUI and delete the Control Function using the handset's menu system.

   For example, do the following on the Nokia 6630 and 6680 handsets to remove the Control Function:

   Select **Menu→Tools→Manager**. Use the arrow keys to select **Netcool** which is the name of the Control Function on the handset. Select **Options→Remove**. You are notified and asked to confirm your intention of removing the component. Select **Yes** to continue.

   During the removal process, a progress bar is displayed. A notification message is presented when the Control Function has been removed.

4. Locate the Monitoring unit components on the handset and delete them using the handset's menu system.

Warning: Do not remove any monitors before stopping the Control Function process.

   For example, do the following on the Nokia 6630 and 6680 handsets to locate and remove the files:

   Select **Menu→Tools→Manager**. Use the arrow keys to select **VoiceMonitor**. Select **Options→Remove**. You are notified and asked to confirm your intention of removing the component. Select **Yes** to continue.

   During the removal process, a progress bar is displayed. A notification message is presented when the Voice Monitor has been removed.

   Do the same for the other monitor components.

Tip: If you are re-installing a monitor, or installing a newer version of a monitor, do so at this point. Do not reboot the handset, as instructed in the next step.

5. Reboot the handset.

Chapter 3: Configuration

This chapter describes how to configure the Netcool for Wireless User Quality Monitoring unit components. The configuration values have to be set in an XML configuration file first and then the file has to be downloaded to the handset.

This chapter contains the following sections:

3.1 Configuration File Settings

The `uqmconfig.xml` configuration file holds all the settings required for the Netcool for Wireless User Quality Monitoring unit to operate.

The XML file has several sections, each dedicated to setting a characteristic of the Monitoring unit. The Control Function parses this XML file and reads its values which determine the operation of the Monitoring unit. The following sections guide you through the setting of the XML file broken down according to sections of the file. They discuss the structure of the XML file, providing information on value ranges and example settings.

Warning: To ensure that your configuration file is free from errors, note that all sections in the file are required unless described as optional in this guide.

For a complete XML configuration file example, see Appendix A: *XML File Example* on page 123.

Note: Make sure you use an XML parser to validate the configuration file before downloading it to the handset. Although the Control Function parses the XML configuration file, it cannot provide information on where the error is. If an invalid XML structure is detected by the Control Function, the Netcool for Wireless User Quality Monitoring unit cannot function.

The <cfengine> Root Element

The `<cfengine>` start and end tags form the boundaries of the root element of the Monitoring unit's XML configuration file. These tags enclose all data and child elements that carry configuration values for the Monitoring unit components. The Control Function uses the information between the `<cfengine>` and `</cfengine>` tags to extract all the settings it needs in order to control itself and the monitors.

Example of <cfengine> Structure

```
[1]   <cfengine compatible='1' release='1'>
[2]     <configfetcher>
[3]       ...
[4]     </configfetcher>
[5]     <producer>
[6]       ...
[7]     </producer>
[8]     <iap>
[9]       ...
[10]    </iap>
[11]    <reporter>
[12]      ...
```

```
[13]     </reporter>
[14]   </cfengine>
```

The <cfengine> start tag includes two attributes. Table 2 summarizes the attributes and nested sections within <cfengine>.

Table 2: Content of <cfengine>

Setting	Description	Values
compatible	The compatible attribute relates to the release of the Netcool for Wireless User Quality Monitoring unit components, and ensures a compatible configuration is supplied.	For Netcool for Wireless User Quality release 2.0 this should be set to 1.
release	The release attribute's value is ignored by the Control Function. It is only an aid provided for users to keep track of, and differentiate between, successive changes to their configuration.	A numeric value.
<configfetcher>	The <configfetcher> section define settings that control automatic configuration file updates.	See *The <configfetcher> Section (optional)* on page 43.
<producer>	The <producer> defines settings for starting up monitors, filtering their events, and queueing the events generated by the monitors.	See *The <producer> Section* on page 44.
<iap>	The <iap> section tells the Control Function how to create an Internet Access Point (IAP) to use for event transmission based on the name of the operator that the handset is connected to.	See *The <iap> Section* on page 71.
<reporter>	The <reporter> section define settings for packaging events and transmitting them to the Netcool for Wireless User Quality Core unit's Event Collector.	See *The <reporter> Section* on page 75.

The <configfetcher> Section (optional)

The values within the <configfetcher> tags instruct the Control Function when to download another XML configuration file. You can use the <fetchperiod> tags to ensure the Control Function downloads a configuration file every time the handset is switched on. The following example shows the structure of the <configfetcher> section.

Example of <configfetcher> Section's Structure

```
[1]   <configfetcher>
[2]     <fetchperiod>86400</fetchperiod>
[3]     <maxperiodextensionpercent>10</maxperiodextensionpercent>
[4]     <url>http://wuq.config.com/uqmconfig.xml</url>
[5]   </configfetcher>
```

Table 3 summarizes the settings within the <configfetcher> tags.

Table 3: Content of <configfetcher> Tags

Setting	Description	Values
<fetchperiod>	Setting <fetchperiod> defines the period, in seconds, after the Control Function is started, before the Control Function downloads a new configuration file. The exact timing of the check is defined by increasing fetchperiod by a random percentage. This prevents many handsets accessing the configuration file at the same time. **Note:** The Control Function does not actually check whether the downloaded configuration file is newer than the existing one. The existing file on the handset is overwritten, even if the replacement file is the same.	A numeric value.
<maxperiod extension percent>	The maximum percentage by which the <fetchperiod> is increased. This allows a random delay to be introduced into the timing of the configuration file download. For example: <fetchperiod>10000</fetchperiod> <maxperiodextensionpercent>10</maxperiodextension percent> In the above example, a new file is downloaded at any time between 10000 seconds and 11000 seconds after the Control Function starts.	A numeric value.
<url>	The location of the replacement configuration file.	The unique URL name identifying the server. Alternatively, you can use the IP address of the server.

The <producer> Section

The values within the <producer> tags instruct the Control Function on which monitors should start up, what events it should accept from the monitors, and how large the event queue size can grow. There is also an option for inserting fixed values into the events to carry information added by the user.

The <producer> section's structure is presented in the following example, and the nested sections are described afterwards.

Example of <producer> Section's Structure

```
[1]   <producer compatible='1' release='1'>
[2]     <waitfornet>
[3]     ...
[4]     </waitfornet>
[5]     <monitor>
[6]       ...
```

```
[7]     </monitor>
[8]     <monitor>
[9]       ...
[10]    </monitor>
[11]    <setresults>
[12]      ...
[13]    </setresults>
[14]    <filter>
[15]      ...
[16]    </filter>
[17]    <queue>
[18]      ...
[19]    </queue>
[20]    <positionfinder>
[21]      ...
[22]    </positionfinder>
[23]    <maxageseconds>
[24]      ...
[25]    </maxageseconds>
[26]  </producer>
```

Table 4 summarizes the settings within the `<producer>` tags.

Table 4: Content of <producer> Tags (1 of 2)

Setting	Description	Values
`compatible`	The `compatible` attribute relates to the release of the Netcool for Wireless User Quality Monitoring unit components, and ensures a compatible configuration is supplied.	For Netcool for Wireless User Quality release 2.0, this should be set to `1`.
`release`	The `release` attribute's value is ignored by the Control Function. The attribute is provided for users to keep track of, and differentiate between, successive changes to their configuration.	A numeric value.
`<waitfornet>`	Prevents monitors from loading unless connected to a specific network.	See *The <waitfornet> Section (optional)* on page 46.
`<monitor>`	Holds information about the monitors. Separate sections are included for each monitor type.	See *The <monitor> Sections* on page 47.
`<setresults>`	Provides the option to insert fixed values into events.	See *The <setresults> Section (optional)* on page 64.
`<filter>`	Contains filtering criteria that determine which events are reported or discarded.	See *The <filter> Section (optional)* on page 64.
`<queue>`	Defines the maximum size of the event queue.	See *The <queue> Section* on page 69.

Table 4: Content of <producer> Tags (2 of 2)

Setting	Description	Values
`<positionfinder>`	Contains the connection settings to a GPS device.	See *The <positionfinder> Section (optional)* on page 70.
`<maxageseconds>`	Defines the maximum age of events in seconds.	See *The <maxageseconds> Section (optional)* on page 71.

The <waitfornet> Section (optional)

There are certain situations where it is not appropriate for monitoring to start. For example, a network provider may not require reports to be sent if a user has taken their handset abroad. Also, if a user changes to a different network provider and changes the SIM in their handset, it may not be appropriate for the monitors to continue operating on the handset.

These situations can be handled properly by waiting for connection to a specific network before allowing monitoring to start. The `<waitfornet>` section provides control over whether monitoring starts, based on the network the handset is connected to and its location.

You can specify network codes or country codes, or both, which the Control Function accepts or rejects, in evaluating whether to load the monitors. For example, in line 2 of the following example, the Control Function will not initiate monitoring for connections made to the network having a MNC of 99, in the country having an MCC of 234 (UK). However, monitoring will start if a connection is made to any other UK network, as specified in line 3 of the example.

In general, you can specify that all networks for a particular country are accepted, by specifying a country only, as in line 3.

Note: If you do not include the `<waitfornet>` section in the configuration file, or if you do not specify values for `country` and `network`, the Control Function starts the monitors without regard to the network or location.

The following example shows the structure of the `<waitfornet>` section.

Example of <waitfornet> Section's Structure

```
[1]   <waitfornet checkintervalseconds='20' retries='2'>
[2]     <net country='234' network='99'>reject</net>
[3]     <net country='234'>accept</net.
[4]   </waitfornet>
```

Table 5 summarizes the settings within the <waitfornet> tags.

Table 5: Content of <waitfornet> Tags

Setting	Description	Values
checkintervalseconds	The interval, in seconds, between successive checks by the Control Function for connection to a suitable network.	A numeric value, where checkintervalseconds is greater than 0.
retries	The number of times the handset rechecks for connection to a suitable network. Failure to connect results in a 'failed configuration' message.	A numeric value, where retries is greater than or equal to 0.
country	The Mobile Country Code (MCC). Although usually a numeric value, this is a string value because it may not be numeric on other phone systems.	The string value of MCC.
network	The Mobile Network Code (MNC). Although usually a numeric value, this is a string value because it may not be numeric on other phone systems.	The string value of MNC.

The <monitor> Sections

The <monitor> sections tell the Control Function which monitors to load and start. The start-up settings for each monitor to be launched are defined within the <monitor> sections, which are nested within the <producer> section. A<monitor> section is required for each monitor type.

Warning: If the settings for a monitor are not included in the configuration file, then the monitor does not start up, even if it is installed.

The <monitor> section's structure is presented in the following example, and the configuration settings required for each monitor type are described in the sections that follow.

Example of <monitor> Section's Structure

```
[1]   <monitor type='voicemonitor' sysid='271000008' compatible='1' release='1'>
[2]     <numsamples>5</numsamples>
[3]     <samplinginterval>6</samplinginterval>
[4]     <mode>short</mode>
[5]   </monitor>
[6]   <monitor type='datamonitor' sysid='271000010' compatible='1' release='1'>
[7]     <timeout>120</timeout>
[8]     <config label='corporate web site'>
[9]       <ipaddress>10.1.2.3</ipaddress>
[10]      <netmask>255.255.255.255</netmask>
[11]      <rootpage>index.html</rootpage>
[12]      <filename>*.gif</filename>
[13]      <hostname>www.corporatesite-1.com</hostname>
```

```
[14]     </config>
[15]  </monitor>
```

Common Monitor Settings

The <monitor> sections have common settings required by the Control Function for all monitors. These settings are defined as attributes within each <monitor> start tag. Table 6 summarizes the common settings for the monitors. These settings are always required, regardless of the monitor type.

Table 6: Common Settings for Monitors

Common Setting	Description	Values
type	The type of the monitor.	The voicemonitor value designates the Voice Monitor, while the datamonitor value designates that the monitor type is Data. Values for other monitors are: • videomonitor • activeclient
sysid	The unique identification of the monitor on the handset's operating system. The sysid value must be correct, otherwise the wrong monitor might be loaded, and the attempt to run the monitor fails.	On Symbian handsets, the sysid of the monitors is as follows: • Voice Monitor: 271000008 • Data Monitor: 271000010 • Video Monitor: 536877773 • Active Client: 536877768
compatible	The compatible attribute relates to the release of the Netcool for Wireless User Quality Monitoring unit components, and ensures a compatible configuration is supplied.	For Netcool for Wireless User Quality release 2.0 this should be set to 1.
release	This is ignored by the monitor but allows you to assign version numbers to successive changes in the monitor configuration.	A numeric value.

Voice Monitor Settings

In addition to the common settings described in *Common Monitor Settings*, the Voice Monitor requires further values to be defined in order to function properly. These settings determine how the Voice Monitor measures the voice quality of a voice call.

The Voice Monitor has three analysis modes: Short, Full and Continuous. The configuration for each mode is described, using examples, in the following sections.

For further information on the Voice Monitor settings used in these examples, see Table 7 on page 51.

Example of a Short Mode Configuration

```
[1]   <monitor type='voicemonitor' sysid='271000008' compatible='1' release='1'>
[2]     <numsamples>5</numsamples>
[3]     <samplinginterval>6</samplinginterval>
[4]     <mode>short</mode>
[5]   </monitor>
```

The Short analysis mode provides a fast, real-time view of the call quality by only measuring the first section of the call. Two configuration parameters are used to set the Short analysis mode:

- The sampling interval, determined by the `<samplinginterval>` value.
- The number of samples, determined by the `<numsamples>` value.

In Short analysis mode only the first `X` seconds of a voice call is monitored, where:

```
X=<samplinginterval>*<numsamples>
```

Note that `X` should be approximately 30 seconds in order to obtain information fast on the quality of the voice call.

After a call is set up, the Voice Monitor starts sampling the downlink voice waveform immediately, recording successive samples each containing the amount of seconds of speech specified in `<samplinginterval>`. After collecting the number of samples defined in `<numsamples>`, the Voice Monitor computes a valid MOS value for the call and returns an event to the Netcool for Wireless User Quality Core unit during the active call.

Note: The value of the `<samplinginterval>` parameter is used by the Psytechnics NiQA algorithm and should be set to the recommended value of `6`. In order to compute a valid MOS value, a minimum number of five or more samples is required. It is recommended to set the `<numsamples>` value to `5`.

For a summary of the settings used to configure the Voice Monitor, see Table 7 on page 51.

Example of a Full Mode Configuration

```
[1]   <monitor type='voicemonitor' sysid='271000008' compatible='1' release='1'>
[2]     <samplinginterval>6</samplinginterval>
[3]     <mode>full</mode>
[4]   </monitor>
```

As the Voice Monitor in Full analysis mode analyzes the whole length of the call, the `<numsamples>` tag becomes irrelevant, and does not need to be used. In Full analysis mode, the samples are taken at defined regular intervals set by the `<samplinginterval>` tags, and then the results are averaged and reported to the Core unit when the call ends.

For a summary of the settings used to configure the Voice Monitor, see Table 7 on page 51.

Example of a Continuous Mode Configuration

```
[1]   <monitor type='voicemonitor' sysid='271000008' compatible='1' release='1'>
[2]     <numsamples>5</numsamples>
[3]     <samplinginterval>6</samplinginterval>
[4]     <heartbeat>5</heartbeat>
[5]     <mode>continuous</mode>
[6]   </monitor>
```

The Continuous analysis mode is similar to Short analysis mode, but instead of transmitting a single result to the Control Function, it transmits results periodically throughout the call. Two configuration parameters are used to set the Continuous analysis mode:

- The sampling interval, determined by the <samplinginterval> value.
- The number of samples in subsequent events. This indicates how often a result is returned, and is determined by the <heartbeat> value.

As with the Short analysis mode, the first X seconds of a voice call are monitored, where:

```
X=<samplingintervals>*<heartbeat>
```

Note that X should be approximately 30 seconds in order to obtain information fast on the quality of the voice call.

To provide periodic results, the call continues to be monitored, using <heartbeat> to define the number of samples in further events. For example, the next Y seconds of the call is monitored, where:

```
Y=<samplinginterval>*<heartbeat>
```

After the next Y seconds are monitored, another Y seconds are monitored. This is repeated until the end of the call. For example, to generate an event every 30 seconds you set <heartbeat> to 5 and <samplinginterval> to 6.

Note: You must set <heartbeat> greater than or equal to <numsamples>. This means that sampling periods (Y) monitored later in the call must be at least as long as the first sampling period (X).

As in Short analysis mode, the value of the <samplinginterval> parameter is used by the Psytechnics NiQA algorithm and should be set to the recommended value of six. Similarly, in order to compute a valid MOS value, a minimum number of five or more samples is required. The recommended value for <numsamples> is five, and <heartbeat> must be greater than or equal to this, as noted above.

See Table 7 for a summary of the settings used to configure the Voice Monitor.

Table 7: Settings for the Voice Monitor

Voice Monitor Setting	Description	Values
<numsamples>	The number of samples required before a measurement can be considered valid. **Note:** This parameter is only used in Short analysis mode.	A numeric value. The default and recommended value is 5.
<samplinginterval>	The length of each sample taken by the monitor in seconds.	A value in seconds. The default and recommended value is 6.
<heartbeat>	The number of samples in subsequent events. This indicates how often a result shall be returned after the initial X=<samplinginterval>*<numsamples> measurement. **Note:** This parameter is only used in Continuous analysis mode.	A numeric value. The recommended value for <numsamples> is 5, and <heartbeat> must be greater than or equal to this.
<mode>	The mode can be either Short, Full, or Continuous In Short analysis mode, the Voice Monitor measures the first 30 seconds of a call and sends the results during the active call (requires a class A handset) or immediately after the call terminates. **Note:** The <samplinginterval> in Short analysis mode has to be set so that at least five samples can be taken, as five samples are required for a valid measurement. In order to obtain a fast, real-time report on the quality of the voice call, the measurement should monitor approximately the first 30 seconds of a call. Due to this, it is recommended not to set the <samplinginterval> to a higher value than six seconds for Short analysis mode. Also, in some cases when the call lasts less than 30 seconds, there might not be enough samples to generate valid events. In Full analysis mode, the Voice Monitor takes samples during the entire call and then an averaged result for the duration of the entire call is sent after the call terminates. The samples are taken at intervals set by the <samplinginterval> tags. In contrast, the Continuous analysis mode measures the voice quality for the duration of the call and transmits results periodically throughout the call.	short, full, or continuous.

Data Monitor Settings

In addition to the common settings described in *Common Monitor Settings* on page 48, the Data Monitor requires further values to be defined in order to function properly. These settings determine what targets and files the Data Monitor measures in the HTTP transactions initiated from the handset. You can configure the Data Monitor to check the responsiveness and performance of several specific HTTP transactions by defining a `<config>` section for each target and filename you wish to monitor, as described in Table 8 on page 56.

You can define different `<config>` sections for different purposes. The following example configurations are shown in this section:

- Monitoring a single web server
- Monitoring all web servers on a network
- Monitoring for a particular file type
- Monitoring FTP, RTSP, and Multimedia Messaging Service (MMS) transactions
- Monitoring interface problems

For further information on the Data Monitor settings used in these examples, see Table 8 on page 56.

Example of Monitoring a Single Web Server

```
[1]  <monitor type='datamonitor' sysid='271000010' compatible='1' release='1'>
[2]    <timeout>120</timeout>
[3]    <config label='corporate web site'>
[4]      <ipaddress>10.1.2.3</ipaddress>
[5]      <netmask>255.255.255.255</netmask>
[6]      <rootpage>index.html</rootpage>
[7]      <filename>*.gif</filename>
[8]      <hostname>www.corporatesite-1.com</hostname>
[9]    </config>
[10] </monitor>
```

You can monitor a single web server, using the `<ipaddress>`, `<netmask>`, and `<hostname>` tags to define a server and an IP address. Use the netmask value `255.255.255.255` to define `10.1.2.3` as the single IP address that you want to monitor. A logical `AND` operation is performed between an incoming packet's IP address and the value `255.255.255.255`, and if the result matches your defined IP address of `10.1.2.3`, the packet is accepted.

Use the `<hostname>`, `<rootpage>`, and `<filename>` tags to filter visited web pages so you can specify precisely what you want to monitor on the web server. The example configuration shown above instructs the Data Monitor to monitor the server with IP address `10.1.2.3` and generate an event each time a `.gif` file occurs on the `index.html` page of the `www.corporatesite-1.com` web site.

The Data Monitor uses the <hostname>, <rootpage>, and <filename> tags as a means of extra filtering and always checks the values within these tags in the following order:

```
[1]  <hostname>
[2]  <rootpage>
[3]  <filename>
```

For further information on the Data Monitor settings used in these examples, see Table 8 on page 56.

Example of Monitoring all Web Servers on a Network

```
[1]  <monitor type='datamonitor' sysid='271000010' compatible='1' release='1'>
[2]    <timeout>120</timeout>
[3]    <config label='all corporatesite-1 web servers'>
[4]      <ipaddress>192.168.32.0</ipaddress>
[5]      <netmask>255.255.255.0</netmask>
[6]      <rootpage>*</rootpage>
[7]      <filename>*.gif</filename>
[8]      <hostname>*corporatesite-1*</hostname>
[9]    </config>
[10] </monitor>
```

You can also monitor all web servers on a single network, using the <netmask> and the <hostname> tags as described in the previous example, but by monitoring the range of IP addresses 192.168.32.x, where x is any value between 0 and 255. In the preceding example, the value * is contained within the <rootpage> tags. This means that the Data Monitor monitors all pages on the Corporatesite-1 web site and generates an event for every .gif file appearing on the site. You only use the <filename> setting to monitor content within a single web page or a set of web pages specified using the <rootpage> tags.

For further information on the Data Monitor settings used in these examples, see Table 8 on page 56.

Example of Monitoring for a Particular File Type

```
[1]  <monitor type='datamonitor' sysid='271000010' compatible='1' release='1'>
[2]    <timeout>120</timeout>
[3]    <config label='mp3 files only'>
[4]      <ipaddress>0.0.0.0</ipaddress>
[5]      <netmask>0.0.0.0</netmask>
[6]      <rootpage>.mp3</rootpage>
[7]      <filename>*</filename>
[8]      <hostname>*</hostname>
[9]    </config>
[10] </monitor>
```

You can use the Data Monitor to monitor particular file types only. In this example, the <ipaddress> and <netmask> tags contain the value 0.0.0.0, meaning that all IP addresses on the Internet are monitored. The value .mp3 is contained within the <rootpage> tags because in this case .mp3 files are the only content of interest to the Data Monitor. Rather than specifying a set of .html pages, you specify any content with an .mp3 suffix. You then use the value * within the <filename> tags.

For further information on the Data Monitor settings used in these examples, see Table 8 on page 56.

Example of Using the Packet Statistics Feature

```
[1]  <monitor type='datamonitor' sysid='271000010' compatible='1' release='1'>
[2]    <timeout>120</timeout>
[3]    <packetstats enable='true' protocol='any'>
[4]      <poll enable='true'>30</poll>
[5]      <ipfilter>
[6]        <ipaddress>0.0.0.0</ipaddress>
[7]        <netmask>0.0.0.0</netmask>
[8]      </ipfilter>
[9]    </packetstats>
[10]   <config label='world'>
[11]     <ipaddress>0.0.0.0</ipaddress>
[12]     <netmask>0.0.0.0</netmask>
[13]     <rootpage>*</rootpage>
[14]     <filename>*</filename>
[15]     <hostname>*</hostname>
[16]   </config>
[17] </monitor>
```

The above example demonstrates how to enable the Data Monitor's Packet Statistics feature in order to monitor FTP, MMS and RTSP transactions. This optional feature provides statistical information about packet-based services. The example demonstrates how to enable the Packet Statistics feature for both TCP and UDP protocols, using the <packetstats enable='true' protocol='any'> tag. However, you can specify a particular protocol to be monitored, such as TCP, by setting protocol='tcp'.

You can also disable the Packet Statistics feature by setting packetstats enable='false'.

The example also shows how to enable polling, using the <poll> tag. This means that events are based on downloads of at least 30 second duration. See Table 8 on page 56 for more information about the Data Monitor settings.

Example of Using a Filter to Prevent Infinite Loop Behavior

```
[1]  <filter>
[2]    <action>
[3]      <eval>
[4]        <lefthand>token:servicetype</lefthand>
```

```
[5]         <operation>equal</operation>
[6]         <case>
[7]           <value>DATA</value>
[8]           <action>
[9]             <eval>
[10]              <lefthand>token:serverconnection</lefthand>
[11]              <operation>equal</operation>
[12]              <case>
[13]                <value>62.190.48.185:9520</value>
[14]                <action><reject/></action>
[15]              </case>
[16]              <case>
[17]                <value>62.190.48.185:9530</value>
[18]                <action><reject/></action>
[19]              </case>
[20]              <default>
[21]                <action><accept/></action>
[22]              </default>
[23]            </eval>
[24]          </action>
[25]        </case>
[26]        <default>
[27]          <action><accept/></action>
[28]        </default>
[29]      </eval>
[30]    </action>
[31]  </filter>
```

If you enable the Packet Statistics feature and specify a broad range of IP addresses to monitor, the IP address of the Event Collector may be included in the specified range. For example, setting `ipaddress` and `netmask` to `0.0.0.0` specifies a range of IP addresses that includes that of the Event Collector. This results in an 'infinite loop' situation where events sent to the Event Collector are monitored and analyzed by the Data Monitor, causing further events to be created and sent.

In the example above, the Data Monitor Packet Statistics feature does not monitor events destined for the IP address `62.190.48.185`. You can specify the IP address of your Event Collector in a similar filter in order to prevent events being monitored.

Example of Using the Interface Monitoring Feature

```
[1]   <monitor type='datamonitor' sysid='271000010' compatible='1' release='1'>
[2]     <timeout>120</timeout>
[3]     <packetstats enable='false' protocol='any'>
[4]       <poll enable='false'/>
[5]       <ipfilter>
[6]         <ipaddress>0.0.0.0</ipaddress>
[7]         <netmask>0.0.0.0</netmask>
[8]       </ipfilter>
```

```
[9]     </packetstats>
[10]    <ifmonitor enable='true'>
[11]      <apnname>Telephone Internet</apnname>
[12]      <apnname>Contract Internet</apnname>
[13]    </ifmonitor>
[14]    <config label='world'>
[15]      <ipaddress>0.0.0.0</ipaddress>
[16]      <netmask>0.0.0.0</netmask>
[17]      <rootpage>*</rootpage>
[18]      <filename>*</filename>
[19]      <hostname>*</hostname>
[20]    </config>
[21]  </monitor>
```

The Interface Monitoring feature provides information about failed connections. To enable Interface Monitoring, use the `<ifmonitor enable='true'>` tag as shown in the above example. The example demonstrates how to monitor interface problems for a specific APN. You can monitor more than one APN. The example shows how to monitor connections on the `'Telephone Internet'` and `'Contract Internet'` APNs.

In the event of a failed connection, an error code is sent to the Event Collector, identifying the reason for the failure. For successful connections, the time taken to establish the connection is sent to the Event Collector. For more information about these results, see *Interface Monitoring* on page 112.

The Packet Statistics feature is not required in this particular application so it is disabled by setting `packetstats enable='false'`.

See Table 8 for a summary of the settings used to configure the Data Monitor.

Table 8: Settings for the Data Monitor (1 of 4)

Data Monitor Settings	Description	Values
`<timeout>`	The Data Monitor does not monitor transactions lasting longer than the time set in the `<timeout>` tags. This is a *global* setting valid for each configuration of the Data Monitor. The setting controls the usage of the Data Monitor's memory, which has to retain copies of the IP packets in the transaction in order to analyze the load. If a transaction lasts too long, then the memory usage could grow significantly.	A numeric value, in seconds.
`<config>`	The settings within the `<config>` tags hold the instructions for the Data Monitor to monitor the packets that match the specified `ipaddress`, netmask, rootpage, filename and hostname. There may be several configurations defined, each set by its own `<config>` section.	See the row entries below.
`label`	The `label` attribute defines a friendly name for the configuration.	A name provided by the user.

Table 8: Settings for the Data Monitor (2 of 4)

Data Monitor Settings	Description	Values
<ipaddress>	The <ipaddress> tags contain the IP address of the target content server. HTTP transactions involving the specified server are monitored. To define a range of IP addresses, use the <netmask> tag. A logical AND operation is performed between an incoming packet's IP address and the <netmask> tag, and the result is compared with your defined IP address to decide whether to accept or reject the packet.	IP address in dotted decimal format. For example, 212.183.137.12.
<netmask>	Use the <netmask> tag to filter a range of IP addresses of interest. The <netmask> is used in a logical AND operation with the incoming packet's IP address to decide whether the packet originates from the required range of IP addresses. For example, to monitor all data originating from 212.183.137.x, set <netmask> to 255.255.255.0.	The netmask in dotted decimal format. For example, 255.255.255.0.
<hostname>	The <hostname> tags contain the hostname of the target content server. This is the host header field of the HTTP query as defined in RFC 1945. Wildcard characters * and ? can be used to substitute other characters in the match. The asterisk (*) stands for zero or more occurrences of any characters, while the question mark (?) stands for a single occurrence of any character.	The unique name identifying the server. For example, live.corp-2.co.uk.
<filename>	The <filename> tags specify what file types or exact file names should be monitored. Only HTTP transactions of the files matching the specified value are tracked by the Data Monitor. Wildcard characters * and ? can be used to substitute other characters in the match. The asterisk (*) stands for zero or more occurrences of any characters, while the question mark (?) stands for a single occurrence of any character.	A file type or file name. For example, the *.gif value means that the download of GIF images is monitored. Another example is the *.mp3 value, which means that the download of MP3 files is monitored. You can also specify exact file names or HTML page downloads to monitor. In case of HTML page downloads, the path section of the HTTP URL is checked. For example, about.corp-2.html.
<rootpage>	The <rootpage> tags specify a web page that should be monitored. Wildcard characters * and ? can be used to substitute other characters in the match. The asterisk (*) stands for zero or more occurrences of any characters, while the question mark (?) stands for a single occurrence of any character.	A web page URL. For example, http://www.corp-2.com/index.html.

Table 8: Settings for the Data Monitor (3 of 4)

Data Monitor Settings	Description	Values
<packetstats>	The <packetstats> tags allow you to enable or disable the Data Monitor Packet Statistics feature. You can use the Packet Statistics feature to monitor TCP packets only, UDP packets only, or both. Set the enable attribute to true or false as required. For example: <packetstats enable='true' protocol='tcp'>	Set the enable and protocol attributes to one of the string values listed: enable: true or false protocol: tcp, udp, or any **Note:** If you want to monitor FTP transactions, set this to tcp.
<poll>	The <poll> tags allow you to send events at periodic intervals. These tags are useful when monitoring socket connections that send large amounts of data, such as video streams and FTP downloads. Set an interval, in seconds, that must elapse before an event is sent. For example, when monitoring a video stream, if you set polling to 30 seconds, then an event is sent after the stream has been monitored for 30 seconds, and at regular 30 second intervals until the stream ends, or until <timeout> is reached. The above example appears in the configuration file as follows: <poll enable='true'>30</poll>	Set the enable attribute to true or false.
<ipfilter>	The <ipfilter> tags allow you to specify IP ranges to be monitored. Use these tags in conjunction with the <ipaddress> and <netmask> tags. You can use multiple <ipfilter> sections to define different ranges of IP addresses. **Note:** The <ipfilter> tags are designed to permit filtering of IP addresses from the range specified using the <ipaddress> and <netmask> tags in the Data Monitor <config> section. When using <ipfilter>, only specify IP addresses that are a subset of the range specified in the <config> section.	For <ipaddress> and <netmask>, specify IP addresses in dotted decimal format. See the <ipaddress> and <netmask> row entries above for more information.

Table 8: Settings for the Data Monitor (4 of 4)

Data Monitor Settings	Description	Values
`<ifmonitor>`	The `<ifmonitor>` tags allow you to enable or disable the Data Monitor Interface Monitoring feature. You can use the Interface Monitoring feature to obtain information about failed connections. Set the `enable` attribute to `true` or `false` as required. For example, to enable Interface Monitoring: `<ifmonitor enable='true'>`	Set the enable attribute to `true` or `false`.
`<apnname>`	The `<apnname>` tags allow you to specify the Access Point Name (APN) that you want to obtain information about. This name is supplied by the network provider for use when connecting using the provider's service. If the connection fails, the Interface Monitoring feature generates an event that includes the APN name. For example: `<apnname>Contract Internet</apnname>`	A string value, as specified by the provider.

Video Monitor Settings

In addition to the common settings described in *Common Monitor Settings* on page 48, the Video Monitor requires further values to be defined in order to function properly. These settings allow you to specify which video streams are to be monitored, based on the originating IP address of video packets, and also to control how much of a video stream is monitored.

The following example demonstrates how to configure monitoring of a video stream.

Example of Monitoring a Video Stream

```
[1]  <monitor type='videomonitor' sysid='\' compatible='1' release='1'>
[2]    <sampleperiod>3</sampleperiod>
[3]    <config label='all'>
[4]      <ipaddress>10.1.2.3</ipaddress>
[5]      <netmask>255.255.255.255</netmask>
[6]    </config>
[7]  </monitor>
```

The above example shows how to monitor a 3 second video stream from a specific server. Use the netmask value `255.255.255.255` to define `10.1.2.3` as the single IP address that you want to monitor. A logical `AND` operation is performed between an incoming packet's IP address and the value `255.255.255.255`, and if the result matches your defined IP address of 10.1.2.3, the packet is accepted.

See Table 9 for a summary of the settings used to configure the Video Monitor.

Table 9: Settings for the Video Monitor

Video Monitor Settings	Description	Values
`<sampleperiod>`	The length of the sample, in seconds, on which results are based. You can monitor the entire duration of a video stream, or sample only a portion of it. To get results based on the entire duration of a stream, set this to `0`.	A numeric value in seconds.
`<config>`	The settings within the `<config>` tags instruct the Video Monitor to monitor the packets that match the specified `ipaddress` and `netmask`. There may be several configurations defined, each set by its own `<config>` section.	See the row entries below.
`label`	The `label` attribute defines a meaningful name for the configuration.	A string value.
`<ipaddress>`	The `<ipaddress>` tags contain the IP address of the target content server.	IP address in dotted decimal format. For example, `212.183.137.12`.
`<netmask>`	An `AND` operation is performed between the netmask and the IP address of the incoming packet in order to verify it is from the intended server.	Netmask in dotted decimal format. For example, `0.0.0.0`.

Active Client Settings

In addition to the common settings described in *Common Monitor Settings* on page 48, the Active Client requires further values to be defined in order to function properly. These settings allow you to set up sequences of actions that simulate user handset operation. The sequences can consist of any combination of Voice, HTTP, WAP/I-Mode, or RTP/RTSP actions. You can also separate sequences using an inter-execution delay.

The following example shows a typical Active Client configuration containing a single schedule that consists of various types of actions.

Tip: For reconfiguring the Active Client, using automatic configuration downloads is recommended. To download new configuration files automatically, include a `<configfetcher>` section in your configuration file, as described in *Example of <configfetcher> Section's Structure* on page 43. You should also ensure that a current configuration file is always available for download.

Example of Active Client Configuration

```
[1]   <monitor type='activeclient' sysid='536877768' compatible='1' release='1'>
[2]     <splashscreen>on</splashscreen>
[3]     <periodicity>3600</periodicity>
```

```
[4]       <schedule>
[5]         <action type='delay'>
[6]           <interval>30</interval>
[7]         </action>
[8]         <action type='call'>
[9]           <mute>1</mute>
[10]          <phone>121</phone>
[11]        </action>
[12]        <action type='delay'>
[13]          <interval>35</interval>
[14]        </action>
[15]        <action type='hangup'/>
[16]        <action type='delay'>
[17]          <interval>60</interval>
[18]        </action>
[19]        <action type='stream'>
[20]          <uri>rtsp://stream.providerx.co.uk</uri>
[21]          <apn>ProviderX live!</apn>
[22]          <volume>0</volume>
[23]          <duration>60</duration>
[24]        </action>
[25]        <action type='httpget'>
[26]          <uri>http://www.content.com/index.html</uri>
[27]          <apn>AnAPN</apn>
[28]        </action>
[29]      </schedule>
[30]  </monitor>
```

The Active Client is configured to perform the following actions in the example above:

- Wait 30 seconds, mute the handset, then dial the number 121 and wait for 35 seconds before hanging up.
- Wait 60 seconds, then launch a streaming video connection for 60 seconds with muted stream volume.
- Immediately download a Web site using the specified Access Point Name (APN).

While the handset performs the set actions, the monitors can collect various performance metrics. Make sure you have the appropriate monitors installed and configured on the handset in order to receive event information about the actions performed.

In the example above, the Active Client is configured to display a splash screen during the time of the client's operation. The `<periodicity>` tags specify that the sequence of actions is to be repeated 1 hour after the sequence ends.

Note: When the sequence of actions ends, the Active Client repeats the sequence after the time specified in the `<periodicity>` tags elapses. If you wish to stop the Active Client from repeating the sequence, update the XML configuration file with one that does not have settings for the Active Client. The Active Client is not active if the Control Function reads the configuration file and does not find a section for the Active Client.

When the Active Client initiates actions, the following behaviors apply:

- Only one Voice action can run concurrently.
- Only one Video action can run concurrently.
- Voice actions and Data actions (for example, HTTP, WAP/I-Mode, or RTP/RTSP) can run concurrently.
- Only one Data action can run concurrently. For example, if a sequence consists of one Voice and two Data actions, the Voice action and the first Data action can run concurrently, but the second Data action cannot run until the first one has completed.
- A Video action cannot run at the same time as a Data action.

Table 10 summarizes the settings used to configure the Active Client.

Table 10: Settings for the Active Client (1 of 2)

Active Client Settings	Description	Values
`<splashscreen>`	You can display an optional splash screen to indicate to the user that the Active Client is operating.	`On` or `Off`.
`<periodicity>`	Defines the time, in seconds, between a sequence of actions ending and restarting.	A numeric value.
`<schedule>`	The `<schedule>` tags contain all the actions, and intervals between actions, that form part of the sequence. Everything within the `<schedule>` tags is executed in the order shown.	See the row entries below.
`<action>`	The `<action>` tags describe the type of action to be performed.	See the row entries below.

Table 10: Settings for the Active Client (2 of 2)

Active Client Settings	Description	Values
`type`	The type of action. This indicates whether the action opens or closes a voice call, or whether it is a web page interaction, a video interaction, or a delay action.	A string having one of the following values: • `delay`: Specifies a delay between actions, or the length of a voice call; must be followed by the `<interval>` tags (see the row entries below). • `call`: Specifies that a voice call is to be initiated; must be followed by the `<phone>` tags (see below). • `hangup`: Terminates a voice call. • `stream`: Specifies that a video stream is to be initiated; see Table 11 on page 63 for settings. • `httpget`: Fetches a file from an HTTP server.
`<interval>`	Use `<interval>` to insert a delay in the sequence. The delay is in seconds and can be between actions, or within the actions. For example, use a delay within a voice call action to separate the opening and closing of a call.	A numeric value.
`<phone>`	The number to be dialled.	A numeric value.
`<mute>`	Use `<mute>` to switch off the handset's microphone.	`<mute>` can have the following values: • `0`: The handset's microphone is switched on. • `1`: The handset's microphone is switched off.

Table 11 summarizes the settings within the `stream` values.

Table 11: Settings for the <stream> Section (1 of 2)

Setting	Description	Value
`<uri>`	The URL to be accessed in an HTTP action.	A web page URL. For example, `http://www.abcd.com/index.html`.
`<apn>`	The Access Point Name (APN) the Active Client uses when performing an HTTP operation. This name is supplied by the network provider for use when connecting to the internet from the handset, using the provider's service.	A string value, as specified by the provider. For example, `internet`.

Table 11: Settings for the <stream> Section (2 of 2)

Setting	Description	Value
`<volume>`	You can mute the stream volume by setting the `<volume>` tag to 0. For example, `<volume>0</volume>`. If you want to hear sound from the stream, then do not include the `<volume>` tag.	If used, set to `0`.
`<duration>`	Use the `<duration>` tag to specify how long to stream video for. Specify a value in seconds. For example `<duration>60</duration>`	A numeric value.

The <setresults> Section (optional)

The `<setresults>` section provides a way of inserting fixed values into the event. You can insert any 'static' value into an event from by adding this section in the configuration file.

An example of the use of the `<setresults>` section is presented below. In this case, the `customername` result is specified by providing the name of the customer (for example, a company called BigCorp). The name of the customer is inserted into each event coming from monitors deployed on the handsets of the customer. This enables you to determine which customer the event came from.

Example of <setresults> Section's Structure

```
[1]   <setresults>
[2]     <set name='customername'>BigCorp</set>
[3]   </setresults>
```

The `$customername` token is always present in the event results, although it returns a `None` value if it is not set in the `<setresults>` section.

Note: If a fixed value is set in the `<setresults>` section, a token named after the attribute of the `<set>` tag is created to carry that set value in the event.

The <filter> Section (optional)

The `<filter>` section determines which events to accept or discard. Filters can be used to limit network load by selectively discarding events according to their type and to results.

The Control Function passes the events through the set filters and decides to accept or reject them. If an event is accepted based on the filter criteria, it is placed into the queue. The Control Function pulls events from the queue for transmission to the Event Collector according to the settings defined in the `<reporter>` section, as described in *The <reporter> Section* on page 75.

You can set several filters, all of which serve different purposes. The `<filter>` section contains the `<action>` tags which hold the instructions for filtering and can have the following values:

- The `<accept/>` value allows the event to be placed in the queue to be sent.
- The `<reject/>` value discards the event.
- An `<eval>` section contains a condition according to which the event must be evaluated before the Control Function decides on whether to accept or reject it. See *Using <eval> to Make Decisions* on page 66 for details on how to use the `<eval>` section.

The following sections discuss the filtering options with a series of examples to provide real-life usage scenarios.

The Pass Through Filter

The *pass through* filter is the most simple filter as it accepts all events from each monitor and allows them to be placed in the queue for transmission.

Example of Pass Through Filter

```
[1]  <filter>
[2]    <action>
[3]      <accept/>
[4]    </action>
[5]  </filter>
```

Using <eval> to Make Decisions

The <eval> tags hold the filtering criteria that define whether the Control Function accepts or discards an event. Table 12 summarizes the settings for the <eval> section.

Table 12: Content of <eval> Tags

Setting	Description	Value
<lefthand>	The <lefthand> tags hold the name of the event result or Key Performance Indicator (KPI) you wish to compare values against. In other words, this particular data of the event is compared against defined values.	Typically an event result specified by the token name that carries its value. For example, token:servicetype means the type of the monitor that generated the event.
<operation>	The <operation> tags tell the Control Function what comparison to apply to the <lefthand> value.	Comparison options include: • equal: The <action> of the <case> is performed if the result or KPI specified in <lefthand> is equal to the string or number in the case <value> tags, as described in the examples in *Building Filters* on page 66. • gte: If the <lefthand> value is greater than or equal to the case <value>. • lte: If the <lefthand> value is less than or equal to the case <value>. • notequal: If the <lefthand> value is not equal to the case <value>.
<case>	The <case> tags contain the value statements against which you wish to compare the <lefthand> value.	Contains a <value> tag for inserting the exact string or number to compare against, and an <action> tag that determines the action taken if the comparison is matched. The <action> value can be <accept/>, <reject/>, or another <eval> section.
<default>	The <default> tags tell the Control Function what to do if the event did not meet any of the conditions set by the comparison. In other words, this is the 'otherwise' action to perform when none of the conditions are met.	The <default> tags have to contain an <action> tag which can have the value of <accept/>, <reject/>, or another <eval> section.

Building Filters

The following sections guide you through the process of building a filter. Several <eval> sections can be included in the filter, creating customized conditions for accepting or rejecting events. The examples demonstrate only a possible classification of events according to the monitor that generated them and at least one KPI, such as the MOS of a Voice Monitor event.

The following example shows a simple filter that determines whether the event has come from a Voice or a Data Monitor, and only accepts events from the Voice Monitor.

Example of Simple Filter

```
[1]  <filter>
[2]    <action>
[3]    <eval>
[4]      <lefthand>token:servicetype</lefthand>
[5]      <operation>equal</operation>
[6]      <case>
[7]        <value>VOICE</value>
[8]        <action>
[9]          <accept/>
[10]       </action>
[11]     </case>
[12]     <default>
[13]       <action>
[14]         <reject/>
[15]       </action>
[16]     </default>
[17]   </eval>
[18]   </action>
[19] </filter>
```

The settings above instruct the Control Function to take the `servicetype` token's value in the event and perform an 'equal to' comparison operation, comparing it to the value between the `<case>` tags. If the `servicetype` (the monitor type) of the event is `VOICE`, then it equals the value of the `<case>` tags, and the Control Function accepts the event from the monitor. Otherwise, the event is discarded according to the `<default>` tag settings.

The following example shows a filter setting that only reports 'bad' voice calls based on the Mean Opinion Score (MOS) value of the Voice Monitor event.

Example of Filter for Bad Voice Calls

```
[1]  <filter>
[2]    <action>
[3]    <eval>
[4]      <lefthand>token:mos</lefthand>
[5]      <operation>lte</operation>
[6]      <case>
[7]        <value>3.5</value>
[8]        <action>
[9]          <accept/>
[10]       </action>
[11]     </case>
[12]     <default>
```

```
[13]        <action>
[14]          <reject/>
[15]        </action>
[16]      </default>
[17]    </eval>
[18]    </action>
[19]  </filter>
```

The filter settings above instruct the Control Function to examine the `mos` token's value in the event and only accept the event if this value is less than or equal to `3.5` (MOS scores under this value are considered an inadequate voice quality level.) Otherwise, the event is discarded.

Note: In this case, all events from the Data Monitor are rejected, as only events from the Voice Monitor have MOS values.

While the filter above is good for reporting only bad performing voice call data, it rejects all events from the Data Monitor. To set the filter so that it accepts events from the Data Monitor as well, you need to insert the filter for bad voice calls into the previously discussed filter which only accepts events from the Voice Monitor. This way, only voice events are checked for MOS values. Also, a `<case>` tag with the value `DATA` has to be added to the filter. The following example demonstrates this setting.

Example of Filter for Bad Voice Call and Data Events

```
[1]   <filter>
[2]     <action>
[3]     <eval>
[4]       <lefthand>token:servicetype</lefthand>
[5]       <operation>equal</operation>
[6]       <case>
[7]         <value>VOICE</value>
[8]         <action>
[9]         <eval>
[10]          <lefthand>token:mos</lefthand>
[11]          <operation>lte</operation>
[12]          <case>
[13]            <value>3.5</value>
[14]            <action>
[15]              <accept/>
[16]            </action>
[17]          </case>
[18]          <default>
[19]            <action>
[20]              <reject/>
[21]            </action>
[22]          </default>
[23]        </eval>
[24]        </action>
```

```
[25]       </case>
[26]       <case>
[27]         <value>DATA</value>
[28]         <action>
[29]           <accept/>
[30]         </action>
[31]       </case>
[32]       <default>
[33]         <action>
[34]           <reject/>
[35]         </action>
[36]       </default>
[37]     </eval>
[38]     </action>
[39]  </filter>
```

In the example above, instead of an `<accept/>` action value for each voice event, the `<eval>` section discussed in the filter for bad voice has been inserted. This `<eval>` section tells the Control Function to check MOS values of Voice Monitor events only, and accept those with a value under `3.5`. Another `<case>` section has been added with instructions to also accept the event if it had a `DATA` value in its `servicetype` token. As a result, the above settings create a filter that instructs the Control Function to accept any event from the Voice Monitor with an MOS value less than or equal to `3.5` and also to accept any event from the Data Monitor.

For a further example of an XML configuration file, see *XML File Example* on page 123.

Note: When building filters, make sure you are providing correct values and structure. Also, make sure you use an XML parser to validate the configuration file after you have finished providing the settings.

The <queue> Section

The `<queue>` section configures the event queue used to store events prior to transmission. For example, if a connection is temporarily unavailable, then the events are not lost but stored in the queue. There is a limit on the number of events that can be stored. This limit is specified by the contents of the `<size>` tag. When the number of events in the queue exceeds the set limit, the oldest event is discarded.

Also, events are removed from the queue for transmission in the 'oldest first' order. However, if a transmission fails, the events that could not be sent are placed in the queue again. The events are again ordered so that the oldest is sent first in the next transmission.

Adding a `<queue>` section to the configuration file is especially useful in certain cases. For example, if the handset is not capable of simultaneous voice and data transmissions (not a GPRS class A device or a 3G handset), then the Control Function must queue events from the Voice Monitor from the start of the call until the call terminates.

In the following example, the maximum size of the queue is set to 10 events. This means that if a new event arrives in the queue when there are already 10 events being held, then the oldest event is discarded and the new event is added.

Example of <queue> Section's Structure

```
[1]   <queue>
[2]     <size>10</size>
[3]   </queue>
```

Table 13 summarizes the settings within the `<queue>` tags.

Table 13: Content of <queue> Tags

Setting	Description	Values
`<size>`	Specifies the number of events to hold in the queue. If events cannot be transmitted, then they are queued up to this limit. If the number of events in the queue rises above the set size after an event is added, then the oldest event in the queue is discarded.	An integer value. The valid range is `1` to `1000`.

The <positionfinder> Section (optional)

The Control Function can add GPS location information to events if the XML configuration file contains the `<positionfinder>` section, which specifies the connection to a GPS device. The `<positionfinder>` section contains settings for connecting to a GPS device through Bluetooth.

Note: Any GPS device with Bluetooth capability and support of the NMEA protocol can be connected.

If the GPS device is not available for any reason, then the Control Function attempts to reconnect at 10 second intervals.

The `<positionfinder>` section is shown in the following example.

Example of <positionfinder> Section's Structure

```
[1]   <positionfinder>
[2]     <btport>1</btport>
[3]     <btname>BT-GPS-31829</btname>
[4]     <btname>tomtom</btname>
[5]   </positionfinder>
```

Table 14 summarizes the settings within the <positionfinder> tags.

Table 14: Content of <positionfinder> Tags

Setting	Description	Values
<btport>	The Bluetooth port or channel number the Control Function connects to on the GPS device.	A numeric value greater than 0.
<btname>	The GPS device's Bluetooth name. Bluetooth device names are not unique but in certain situations the name can be used to connect to any available GPS device without needing to specify a different configuration file for each handset. Since a specific GPS device model will have a standard name, the Control Function can connect to the first device it finds with the specified name. **Note:** You can specify more than one <btname>. The first device with a name matching in the list will be used.	A string value. Wild cards * and ? may be used in names.

The <maxageseconds> Section (optional)

The <maxageseconds> section defines a maximum age for the events in seconds. Any event that is older than the time specified in this section (counting from the time it was created) will be discarded. This means the Control Function will not transmit any event that is older than the time specified here.

The following example means that each event has a life of no more than 3600 seconds (1 hour) in the Control Function's queue. Any event older than this in the queue is not sent and is discarded.

Example of <maxageseconds> Section

```
<maxageseconds>3600</maxageseconds>
```

The <iap> Section

The <iap> section tells the Control Function how to create an Internet Access Point (IAP) to use for event transmission based on the name of the operator that the handset is connected to. The <iap> settings are required in reporting events and are also useful in case of roaming, where the handset may connect to different networks through agreements between wireless service providers. These settings allow the handset to report the monitoring events even while roaming in any of several other defined operator networks.

Note: If the handset is attached to a network that is not listed in this section, then the Control Function cannot transmit results back to the Event Collector. It continues to queue the events, although the events are eventually discarded based on the size limit and age settings.

The <iap> section's structure is presented in the following example, and the parts of the section are described afterwards.

Example of <iap> Section's Structure

```
[1]  <iap source='commprefselect'>
[2]    <preconditions></preconditions>
[3]    <commprefselector compatible='1' release='1'>
[4]      ...
[5]    </commprefselector>
[6]  </iap>
```

Table 15 summarizes the settings of the <iap> tags.

Table 15: Content of <iap> Section

Setting	Description	Values
source	The source of the internet connection parameters. The value of <source> tells the Control Function where to read the connection parameters from. The parameters are either contained in the <commprefselector> section of the configuration file (source='commprefselector'), or from an IAP called name, which is already configured on the handset (source='name').	A string value. For example, commprefselector or name, where name is a preconfigured internet access point set on the handset by the service provider. This could be the service provider's general IAP setting which the handset automatically uses to connect to the internet.
<preconditions>	Defines when the internet access point is allowed to make a connection.	See *The <preconditions> Section (optional)* on page 73.
<commprefselector>	The <commprefselector> tag contains information for choosing the APN details appropriate for a particular country.	See *The <commprefselector> Section* on page 73.

Note: If the IAP settings are incorrectly defined, the Control Function may be unable to report any events to the Event Collector. The Control Function buffers the events in the queue up to the defined maximum number and tries to resend them at the specified intervals as described in *The <queue> Section* on page 69. It is recommended that a maximum age applicable to *all* events is set. See *The <maxageseconds> Section (optional)* on page 71 for details. Alternatively, you can also prevent monitors from starting until a suitable network connection is found, using the <waitfornet> section. See *The <waitfornet> Section (optional)* on page 46 for further information.

The <preconditions> Section (optional)

The `<preconditions>` section allows you to define when the internet access point is allowed to make a connection. An example of how the `<preconditions>` section is used is as follows:

Example of <preconditions> Section

```
<preconditions>exclusive</preconditions>
```

The `<preconditions>` tags contain a string value that can be set to the following values:

- `exclusive` - Allows an internet connection to be made only if there are currently no other connections open. Use this value if the network is a GSM network with GPRS capability.
- `exclusive3g` - Allows an internet connection to be made only if there are currently no other connections open. Use this value for a UMTS or CDMA2000 network.

If you do *not* include the `<preconditions>` tags, the Control Function attempts to start its own internet connection regardless of whether other connections are open on the handset. In this case, the Control Function acts depending on whether the network allows multihomed connections and the settings of the `<iptransmitter>` tags:

- If your network allows multihomed connections where one handset can have more than one APN connection open at the same time, the Control Function opens a second APN connection to send events.
- If your network does not allow multihomed connections, the Control Function backs off based on the settings of the `<iptransmitter>` tags (see *The <iptransmitter> Section* on page 76) and waits for the existing APN connection to end in order to use one for reporting.

The <commprefselector> Section

The `<commprefselector>` section contains information for choosing the APN details appropriate for a particular country. You can specify details for numerous APNs, including one to use when fetching a new configuration file. If the handset is used for roaming, the Control Function selects the appropriate APN based on the country the handset is in and the network it is using. The `<commprefselector>` section's structure is presented in the following example, and parts of the section are described afterwards.

Example of <commprefselector> Section's Structure

```
[1]   <commprefselector compatible='1' release='1'>
[2]     <operator type='configfetcher' country='234' networkcode='15'>
[3]       <apn>
[4]         <name>internet</name>
[5]         <username>web</username>
[6]         <password>web</password>
```

```
[7]       </apn>
[8]     </operator>
[9]     <operator country='234' network='30'>
[10]      <apn>
[11]        <name>general.mobile-provider.uk</name>
[12]        <username>user</username>
[13]        <password>one</password>
[14]      </apn>
[15]    </operator>
[16]    <operator country='234' network='31'>
[17]      <apn>
[18]        <name>general.mobile-provider.uk</name>
[19]        <username>user</username>
[20]        <password>mobile</password>
[21]      </apn>
[22]    </operator>
[23]    <operator country='234' network='32'>
[24]      <apn>
[25]        <name>general.mobile-provider.uk</name>
[26]        <username>user</username>
[27]        <password>mobile</password>
[28]      </apn>
[29]    </operator>
[30]  </commprefselector>
```

In the above example, the Control Function examines all the `<operator>` tags until it finds country and network values that match the network the handset is connected to. It then uses the APN details specified within that `<operator>` tag. If the Control Function is connecting to fetch a new configuration file, it only examines `<operator>` tags where `type='configfetcher'`.

Table 16 summarizes the settings of the `<commprefselector>` tags.

Table 16: Content of <commprefselector> Section (1 of 2)

Setting	Description	Values
`compatible`	The `compatible` attribute relates to the release of the Netcool for Wireless User Quality Monitoring unit components, and ensures a compatible configuration is supplied.	For Netcool for Wireless User Quality release 2.0 this should be set to `1`.
`release`	This is ignored by the monitor but allows you to assign version numbers to successive changes in the monitor configuration.	A numeric value.
`<operator>`	Defines the country and the network of the operator.	A string value.
`country`	The Mobile Country Code (MCC) of where the handset is located.	A numeric value.
`network`	The Mobile Network Code (MNC) of the network the handset is connected to.	A numeric value.

Table 16: Content of <commprefselector> Section (2 of 2)

Setting	Description	Values
<apn>	The settings for the APN associated with the defined operator long name. APNs are used to define an Internet IAP for connecting the handset to the Internet.	See the row entries below.
<name>	The name of the APN.	A string value. For example, internet or general.providerx.co.uk.
<username>	A user name to access the Internet.	A string value.
<password>	The user's password.	A string value.

The <reporter> Section

The values within the <reporter> tags contain the reporting details the Control Function follows. The <reporter> settings determine where the Control Function sends the events. In Netcool for Wireless User Quality release 2.0, UDP/IP and TCP/IP are the only reporting mechanisms used.

The <reporter> section's structure is presented in the following example, and the parts of the section are described afterwards.

Example of <reporter> Section's Structure

```
[1]   <reporter compatible='1' release='1'>
[2]     <iptransmitter>
[3]       ...
[4]     </iptransmitter>
[5]     <reports>
[6]       ...
[7]     </reports>
[8]   </reporter>
```

Table 17 summarizes the settings within the <reporter> tags.

Table 17: Content of <reporter> Tags

Setting	Description	Values
compatible	The compatible attribute relates to the release of the Netcool for Wireless User Quality Monitoring unit components, and ensures a compatible configuration is supplied.	For Netcool for Wireless User Quality release 2.0 this should be set to 1.
release	The release attribute's value is ignored by the Control Function. It is only an aid provided for users to keep track of, and differentiate between, successive changes to their configuration.	A numeric value.
<iptransmitter>	Holds information about the way connections are made to the Event Collector.	See *The <iptransmitter> Section* on page 76.
<reports>	Provides details for where and how to send events.	See *The <reports> Section* on page 78.

The <iptransmitter> Section

The values within the <iptransmitter> tags allow you to control aspects of the way connections are made when sending reports to the Event Collector.

The <iptransmitter> section's structure is presented in the following example, and the parts of the section are described afterwards.

Example of <iptransmitter> Section's Structure

```
[1]   <iptransmitter>
[2]     <blackoutfactor>0.5</blackoutfactor>
[3]     <blackoutmin>1</blackoutmin>
[4]     <blackoutmax>60</blackoutmax>
[5]     <backofffactor>1.5</backofffactor>
[6]     <backoffmin>3</backoffmin>
[7]     <backoffmax>20</backoffmax>
[8]   </iptransmitter>
```

In the above example, the <blackoutfactor> tag allows you to control the size of the interval between the end of one connection and a new connection attempt. The <backofffactor> tag controls the interval between successive failed connection attempts. These intervals help to preserve handset battery life.

In the example, a new connection attempt is made after an interval which is 0.5 times the length of the previous connection, up to a maximum size of 60 seconds. If the previous connection lasted 30 seconds, then the interval is 15 seconds. If the previous connection lasted 200 seconds, then the interval is set to the <blackoutmax> value of 60 seconds. The minimum interval is 1 second.

If the connection attempt fails, then another attempt is made after an interval of 3 seconds. If this attempt fails also, then the interval is increased by a factor of 1.5. The process continues until a successful connection is made, with the interval increasing by a factor of 1.5 each time, up to a maximum interval of 20 seconds. When the <backoffmax> value of 20 seconds is reached, the Control Function continues to attempt connections at regular intervals of 20 seconds.

Table 18 summarizes the settings within the <iptransmitter> tags. Use these settings to help preserve handset battery life by controlling how frequently connections are attempted.

Table 18: Content of <iptransmitter> Tags (1 of 2)

Setting	Description	Values
<blackoutfactor>	The factor to be used in calculating the time before another connection is attempted. The duration of the last connection is multiplied by <blackoutfactor> to calculate this interval. For example, if the previous connection was open for 10 seconds, and <blackoutfactor> is set to 2, then a new connection is not attempted for at least 20 seconds.	A floating point numeric value, where blackoutfactor is greater than 0. This setting has no default value.
<blackoutmin>	The minimum time, in seconds, to wait before attempting another connection, following a successful connection.	A numeric value, where blackoutmin is greater than or equal to 0. Also, blackoutmax is greater than blackoutmin. This setting has no default value.
<blackoutmax>	The maximum time, in seconds, to wait before attempting another connection, following a successful connection.	A numeric value, where blackoutmax is greater than or equal to 0. Also, blackoutmax is greater than blackoutmin. This setting has no default value.
<backofffactor>	The factor by which to increase the interval between retries after failed connection attempts. This helps to prevent battery power being consumed by many successive connection attempts if there is a problem connecting. For example, if <backoffmin> is set to 10 and <backofffactor> is set to 1.5, then in the event of a failed connection, the Control Function attempts another connection after 15 seconds. If this subsequent attempt fails, then the next attempt is made after 22.5 seconds. You can set the maximum allowable interval using <backoffmax>.	A floating point numeric value, where backofffactor is greater than 0. This setting has no default value.

Table 18: Content of <iptransmitter> Tags (2 of 2)

Setting	Description	Values
<backoffmin>	The minimum time, in seconds, to wait before attempting another connection, in the event of a failed connection.	A numeric value, where backoffmin is greater than or equal to 0. Also, backoffmax is greater than backoffmin. This setting has no default value.
<backoffmax>	The maximum time, in seconds, to wait before attempting another connection, in the event of a failed connection.	A numeric value, where backoffmax is greater than or equal to 0. Also, backoffmax is greater than backoffmin. This setting has no default value.

The <reports> Section

The <reports> tags contain information about reporting events to the Event Collector. The settings tell the Control Function where to send the events when the time for transmitting them to the Netcool for Wireless User Quality Core unit arrives.

The <reports> section's structure is presented in the example, and the parts of the section are described afterwards. The following example displays a possible setting for UDP reporting. You can configure TCP reporting using exactly the same format, but by replacing <ipreport type='udp'> with <ipreport type='tcp'>.

Warning: The Control Function reports events to the Event Collector using either UDP or TCP. The <reports> section can contain either an <ipreport type='udp'> section or an <ipreport type='tcp'> section, but not both. Including both sections in the configuration file results in an invalid configuration.

Example <reports> Section's Structure

```
[1]   <reports>
[2]     <ipreport type='udp'>
[3]       <ipaddress>199.111.182.302</ipaddress>
[4]       <port>9520</port>
[5]       <maxencrypted>1024</maxencrypted>
[6]       <maxconcatenated>4096</maxconcatenated>
[7]     </ipreport>
[8]   </reports>
```

Table 19 summarizes the settings of the <reports> tags.

Table 19: Content of <reports> Section

Setting	Description	Values
<ipreport>	The tags hold all the information for reporting over UDP or TCP.	See the row entries below.
type	The type of transport used to send the report.	The udp value designates the UDP transport and the tcp value designates TCP. This setting has no default value.
<ipaddress>	The IP address of the Event Collector. This is where the reports are sent.	IP address in dotted decimal format. For example, 199.111.182.302. This setting has no default value.
<port>	The port on which the Event Collector is listening for reports of events.	A port number. For example, 9520. This setting has no default value.
<maxencrypted>	The largest permitted size of a report, in bytes, after compression and encryption. This defines the maximum size of a packet.	A numeric value, where 0 is less than maxencrypted, which is less than 16385. This setting has no default value.
<maxconcatenated>	The required buffer size to allow events to be concatenated before compression.	A numeric value, where 0 is less than maxconcatenated, which is less than 16385.. This setting has no default value.

3.2 Packaging the Configuration File

You can use the `makesis` utility to simplify the process of configuring your handset. By using `makesis`, you can use a `.sis` file to place the XML configuration file in the correct location on your handset. This saves you having to download and install the XML configuration file as a separate process after installing the monitors.

The `makesis` utility is open-source software provided with the Symbian SDK. To benefit from the convenience that `makesis` offers, follow the steps described in this section before installing the monitors, as described in *Installing the Netcool for Wireless User Quality Monitoring Unit* on page 34.

Tip: Using the `makesis` utility is optional, but you may find it particularly useful if you are installing the Netcool for Wireless User Quality monitoring unit on many handsets.

The `makesis` utility uses the `uqmconfig.pkg` file, included with Netcool for Wireless User Quality, to build a `.sis` file from your custom configuration file, `uqmconfig.xml`. You then use this `.sis` file to install your XML configuration file onto your handsets. This method saves you from having to install your configuration file manually.

Follow the steps below:

1. Create your `uqmconfig.xml` configuration file as required, as described in *Configuration File Settings* on page 42. Save it in the same directory as the `uqmconfig.pkg` file and the other Monitoring unit installation files that you downloaded from the support site.

Warning: Make sure you name the Netcool for Wireless User Quality configuration file `uqmconfig.xml` if you wish to package it with the `makesis` utility.

2. Obtain a copy of the `makesis` utility. The utility is included in the Symbian Series 60 SDK which you can download from http://www.symbian.com.
3. Open a command prompt window. Switch to the directory containing the `uqmconfig.pkg` and `uqmconfig.xml` files, and run the following command:

   ```
   makesis uqmconfig.pkg
   ```

   This command creates a file called `uqmconfig.sis`.
4. Download `uqmconfig.sis` and the monitor installation files to your handset, as described in step 2 of *Installing the Netcool for Wireless User Quality Monitoring Unit* on page 34.
5. From your handset, run `uqmconfig.sis`, as described in *Installing the Netcool for Wireless User Quality Monitoring Unit* on page 34. This automatically places the XML configuration file in the correct location on your handset.

Note: If you have followed the above steps, you can ignore the procedure described in *Downloading the Configuration File* on page 82, which describes how to download the XML configuration file to your handset.

3.3 Downloading the Configuration File

When you have set all the required configuration values in the XML configuration file, you are ready to download the file to the handsets.

Warning: If you have used the `makesis` utility to package the configuration file as described in *Packaging the Configuration File* on page 80, then go to *Installing the Netcool for Wireless User Quality Monitoring Unit* on page 34 for instructions on downloading and installing the configuration file together with the other Netcool for Wireless User Quality Monitoring unit components.

There are two main methods of downloading the XML file to a handset:

- HTTP pull operation from the handset
- Using a third-party file transfer mechanism

The following sections provide guidelines for these methods.

Note: Make sure you use an XML parser to validate the configuration file before downloading it to the handset. Although the Control Function parses the XML configuration file, it cannot provide information on where the error is. If an invalid XML structure is detected by the Control Function, the Netcool for Wireless User Quality Monitoring unit displays an error message as it cannot function. If a value is incorrect, the Control Function or the monitor affected cannot start.

Once you have downloaded a configuration file, you can specify whether the Control Function automatically checks for configuration updates. For further information, see *The <configfetcher> Section (optional)* on page 43.

HTTP Pull

To download the XML configuration file to the handset using the Netcool for Wireless User Quality Monitoring unit's engineering GUI, follow these steps:

1. Make sure you have prepared and checked the validity of the XML configuration file. Load the XML file to a web server and make a note of its URL.
2. Enter the Netcool for Wireless User Quality engineering GUI on the handset, as described in step 1 of *Uninstalling the Netcool for Wireless User Quality Monitoring Unit* on page 38.

3. Download the file to your handset through a data connection using the engineering GUI you have entered.

   For example, if you are using the Nokia 6630 or 6680 handset, use the following step to set up a connection.

   Go to **Options→Settings→Edit**. The window for adding the XML file's URL and the APN values for the data connection appears. Enter the following data:

   - The XML configuration file's URL in the **Config URL** box. For example, `http://handset.configuration.com/uqmconfig.xml`
   - The APN name in the **APN Name** box. For example, `internet`.
   - The APN username in the **APN Username** box. For example, `web`.
   - The APN password in the **APN Password** box. For example, `web`.

   Close the *Edit* window once you have populated the URL and APN settings.

4. Select **Options→Settings→Fetch** to start the download of the XML file over the defined data connection.

   The XML file is retrieved over the HTTP data connection. Once the file has been downloaded, a notification on the handset tells you whether the transaction has been successful. When the download is complete and successful, the connection is closed. The XML file is automatically saved to the following location with the following name:

   ```
   c:\systems\apps\netcool\uqmconfig.xml
   ```

   The XML file can have any name while being edited and downloaded, as the engineering GUI's process renames it to `uqmconfig.xml`.

   A notification also tells you which monitors have been started after a successful transaction. This depends on the setting in the XML configuration file. If a monitor does not have a valid setting in the configuration file, it is not launched. Also, if the Control Function notices an error when validating the XML file, it does not accept the file and it notifies you of this in a message.

5. After a successful download, it is recommended to reboot the handset.

Warning: If you have installed the `showgui.sis` file to enable access to the engineering GUI, make sure you disable it when monitoring starts up. Follow the steps in *Disabling the Engineering GUI* on page 36.

File Transfer

You may also use a file transfer mechanism to download the XML file to the handset. For example, the XML file can be downloaded to the handsets via Bluetooth, serial connection, or even Over The Air (OTA), using management software delivered with the handset or a separate provisioning system. See your third party software's user guide for guidelines on downloading files onto handsets.

After rebooting your handset, the Control Function is able to detect when a new configuration file has been downloaded, as long as it is saved to the following location with the following name:

```
c:\systems\apps\netcool\uqmconfig.xml
```

The Control Function looks in the above directory and reads the settings from the `uqmconfig.xml` file.

Warning: The configuration settings only take effect if the XML file is saved to the above location with the correct name.

Some file manager applications for handsets allow you to save a downloaded file as `uqmconfig.xml` to the specified location. See the user guides of the file manager application for details.

3.4 Creating an Owner Acceptance Screen

As a service provider, you can set the Netcool for Wireless User Quality Monitoring unit to display an *owner acceptance screen* before the monitor software becomes available for activation. This is useful when you wish to ask your end users to consent to the operation of performance measuring monitors on their handset.

You can customize the legal note and the graphical branding of the acceptance screen. Your subscribers can either accept or reject the agreement after being presented with the legal note. Monitoring is only enabled if the agreement is accepted. Otherwise, it does not start.

The mechanism which provides this feature can be set up as described in the following section *Creating an Owner Acceptance Screen*.

Creating an Owner Acceptance Screen

To set up a customized owner acceptance screen to your Netcool for Wireless User Quality Monitoring unit, do the following:

1. Create the following files:

   - `operator.mbm`: The MultiBitMap file contains any graphic branding you wish to add to the acceptance screen. For example, you can add the company logo to this file.

     The MultiBitMap file can only be created using the **bmconv.exe** imaging control tool included in the Symbian SDK package.

Note: When creating a logo, the recommended width is 100 pixels, while the recommended height is 60 pixels in order to fit the handset's screen.

   - `legal_notes.txt`: This file contains the legal note and agreement message you wish to display to the handset user on the acceptance screen followed by the options to accept or reject it.

     The `legal_notes.txt` is a plain text file to which you can add any legal and agreement message about the performance monitoring. At the end of the message, make sure you ask for consent, as there will be the option of accepting or rejecting the terms.

Warning: Make sure the filenames are exactly as stated above. The names are detected by the Netcool for Wireless User Quality Monitoring software, so they need to be called `operator.mbm` and `legal_notes.txt` exactly.

Note: The files to create the acceptance screens are not provided by default. This is an optional feature which can be set up by the service provider if the legal environment requires them to do so and if the provider has not asked the end user to sign a paper contract to accept performance monitoring.

2. Make sure you have installed the Monitoring unit components as described in 2.2 *Installation* on page 34, or at least the `Netcool.sis` file.
3. Download the files to the following location as described in *Downloading the Configuration File* on page 82:

   `c:\system\apps\netcool`

   The application detects the files and reads them if present.

Warning: Make sure the files are saved to the above location with the correct name.

4. Reboot the handset. The acceptance screen appears in a few seconds.

   The acceptance screen contains the logo image at the top of the white screen and the legal note below with a **Yes** and **No** button at the bottom.

5. The user can either acceptor reject the terms and conditions expressed in the note displayed.

   The user will only see the acceptance screen once. If the user accepts the legal agreement, the monitoring operation starts according to the configuration file settings. If the user declines to accept the legal agreement, then the monitor applications do not start functioning, and the legal note will not be displayed anymore.

   When the user accepts the agreement, a new file called `legal_shown.txt` is created in the installation directory (`c:\systems\apps\netcool`) containing the character `1`. If the user does not accept the agreement, the `legal_shown.txt` file is created with the character `0`.

Note: To enable a new acceptance screen, deactivate the Netcool for Wireless User Quality Monitoring Unit and then remove the `legal_shown.txt` file.

If the `operator.mbm` and `legal_notes.txt` are not in place, or they do not have any content, then the acceptance screen is *not* displayed, and the monitors start according to the configuration.

Note: The `operator.mbm` is optional. If the operator.mbm is not present, but the `legal_notes.txt` is, the acceptance screen is still displayed with the content of the legal_notes.txt file.

Initiating an Owner Acceptance Screen Over The Air

As the Netcool for Wireless User Quality Monitoring unit components can be installed over the air, the customized acceptance screen files can also be downloaded to the handsets using an appropriate device management application. The same rules apply as described above in *Creating an Owner Acceptance Screen* on page 85, which means that the legal agreement is not displayed until the user reboots the handset.

If you wish to check whether a handset has a legal agreement installed, access the engineering GUI as described in step 1 of *Uninstalling the Netcool for Wireless User Quality Monitoring Unit* on page 38, and go to **Options** and select **Legal**. If the acceptance screen has been set up by creating and downloading the `legal_notes.txt` and `operator.mbm` (optional) files, then the content of the `legal_notes.txt` is displayed. No information is shown if the files are not on the handset in the proper directory.

Chapter 4: Monitoring Event Information

This chapter describes the measurements provided by the Netcool for Wireless User Quality Monitoring unit and the different monitor types. The event data collected is carried by tokens, and the following sections list which token carries what result, as well as detailing what the different value mapping is in the Active Event List (AEL) fields by default.

Certain measured event data, such as timestamp information and event severity level are included in all events, regardless of which monitor generated the event. Other data relates to a particular monitor, and is included in events generated in all modes of operation for that monitor. Other data is unique to a particular monitor when running in a specific mode of operation. All of these measurements are included in this chapter.

This chapter contains the following sections:

4.1 Control Function Measurements

The Control Function detects and places information about the handset and the customer into each event. The measurement results in the events are carried by tokens. Each token's value is mapped to one or more fields in the ObjectServer rules file, and the fields are presented in the columns of the Netcool for Wireless User Quality home page's Active Event List (AEL). This way the event information is presented in columns based on the mapping between the tokens and the fields.

Note: The following tables provide default field mappings and AEL column names. The AEL can be customized using the View Builder features of Netcool/Webtop. For details about customizing the column names and views in Netcool/Webtop, see the *Netcool/Webtop Administration Guide*.

Table 20 lists the data gathered by the Control Function together with information on which token value is assigned to which field, and which AEL columns include the results by default.

Table 20: Information Gathered by the Control Function (1 of 2)

Information	Description	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
IMEI	The handset's International Mobile Equipment Identifier (IMEI).	$imei	`@NodeAlias`	**NodeAlias**; also included in **Identifier** value.
IMSI	The SIM card's International Mobile Subscriber Identity (IMSI).	$imsi	`@Node`; also part of `@Identifier`	**IMSI**; also included in **Identifier** value.
Handset's OS	The handset's operating system (for example, Symbian).	$sysos	Part of `@Agent`	Included in **Agent** value.
OS version	The version of the handset's operating system (for example, 1.2).	$sysver	Part of `@Agent`	Included in **Agent** value.
Handset manufacturer	The name of the handset manufacturer (for example, Nokia).	$manufacturer	Part of `@MobileDevice`	Included in **Mobile Device** value.
Handset model	The model of the handset the monitor is running on. This relies on the settings of the mobile device (for example, 6680).	$model	Part of `@MobileDevice`	Included in **Mobile Device** value.
Customer	The name of the customer set in the `<setresults>` section of the XML configuration file. For example, the name of an enterprise customer to whom the handset belongs. This value is useful for grouping events.	$customername	`@Customer`	Included in **Mobile Device** value.

Table 20: Information Gathered by the Control Function (2 of 2)

Information	Description	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Longitude	GPS location data. The angular distance on the earth's surface, measured in degrees East of the prime meridian (from 0 to 360).	$longitude	`@Location`	Included in **Location** value.
Latitude	GPS location data. The angular distance North of the earth's equator, measured in degrees along a meridian (from 0 to 360).	$latitude	`@Location`	Included in **Location** value.

Note: The information provided by the Control Function is part of every event from the handset, except the GPS location data, which are subject to GPS configuration settings and the GPS device being present and connected. If the GPS device is not available, has not had time to calculate a valid position, or has lost its satellite signal, then the Control Function does not add `$longitude` and `$latitude` tokens to the event. See *The <positionfinder> Section (optional)* on page 70 for more information.

4.2 Measurements Common to all Monitors

Some event data, such as timestamp information and event severity level is included in all events, regardless of which monitor generated the event, or of the operating mode or feature being used. These common measurements are described in this section.

Table 21 shows which results are common to all monitors in all operating modes.

Table 21: Event Results Common to all Monitors (1 of 2)

Information	Description	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Time	The time (UTC time) when the event was generated.	$timestamp	`@LastOccurrence`, `@FirstOccurrence`, `@UQMValue30`; also part of `@Identifier`	**LastOccurrence**; also included in **Identifier** value.
Severity	The level of severity according to the Control Function's calculations.	$severity	`@Severity`	**Severity**
Monitor Type	The type of the monitor which generated the event. The following values apply: • VOICE • DATA • VIDEO	$servicetype	`@Service`; also part of `@Summary`, `@Agent`, and `@Identifier`	**Service**; also included in **Status**, **Agent**, and **Identifier** values.
Bearer	The wireless network bearer (for example, GSM, UMTS, CDMA, or GPRS).	$bearer	`@Bearer`	**Bearer**
Signal Strength	The current signal strength at the end of the event in dBm (dB-milliwatt) which is a logarithmic measurement of signal strength.	$signalstrength	`@SignalStrength`	**SignalStrength**
Cell ID	The identifier of the cell the handset was in when the event was recorded.	$cellid	Part of `@CellID`	**Cell ID**; also included in **Identifier** value.
Location area code	The location area code.	$lac	`@AlertKey`	**Area Code**
Network code	The network code of the operator network.	$networkcode	`@networkcode`	**Network Code**
Country code	The unique country code for an operator network in a country.	$countrycode	`@countrycode`	**Country Code**

Table 21: Event Results Common to all Monitors (2 of 2)

Information	Description	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Type	The application that generated the event. For UQM 2.0, this value is always `UQM`.	$type	`@UQMValue27`	Type
Event filtered	Indicates whether filtering has been applied to the event. This is a boolean value: • `0`: No filtering applied. • `1`: Filtering has been applied.	$filtered	@UQMValue28	Filtered
Timeshot	The time (local time) when the event was generated.	$timeshot	@UQMValue31	Handset Time
UTC offset	The offset, in hours, from UTC time. This value is used, on certain handsets, to calculate the local time. This can be a positive or negative value. For example, a value of -5 represents a time of 5 hours before the current UTC time.	$utcoffset	@UQMValue32	**UTC Offset**
Daylight	A binary value that indicates whether daylight saving time on the handset is switched on or off. The following values apply: • 0: Daylight saving time is off. • 1: Daylight saving time is on.	$daylight	@UQMValue33	Daylight
Time sent	The time (UTC time) when the Control Function sent the event.	$timesent	@UQMValue34	Time Sent
Time received	The time (UTC time) when the Event Collector received the event.	$timereceived	@UQMValue35	Time Received

4.3 Voice Monitor Measurements

The results reported by the Voice Monitor depend on whether it is set to Short, Full or Continuous analysis mode. Results of voice monitoring operations are carried by different tokens using Full mode than when using Short or Continuous mode. However, many tokens map to the same fields in the ObjectServer rules file and the same columns in the AEL, although they contain different measurements.

This section provides details on:

- Measurements in Short and Continuous Modes
- Measurements in Full Mode

For further information on the different operating modes, see *About the Voice Monitor* on page 19.

Note: The tables in the following sections provide default field mappings and AEL column names. The AEL can be customized using the view builder features of Netcool/Webtop. For details about customizing the column names and views in Netcool/Webtop, see the *Netcool/Webtop Administration Guide*.

Measurements in Short and Continuous Modes

Results reported in events are the same for the Short and Continuous voice monitoring modes. These results are shown in Table 22. The table also presents how token values are mapped to the fields, and which AEL columns include the results by default.

Note: Table 22 does not include event data common to all monitors. This is listed in Table 21 on page 92.

Table 22: Event Results in Voice Monitor Short and Continuous Modes (1 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Speech Level	The active speech level of the speech stream during the call. The speech level should be between -16 dboV and -26 dboV for acceptable speech strength. Values range from 0 to -127 dboV.	$speechlevel	`@UQMValue0`	**Speech Level**
Noise Level	The noise level of the speech stream during the call. The lower the noise level, the better the possible listening quality. An acceptable noise level is -40 dboV. Values range from 0 to -127 dboV.	$noiselevel	`@UQMValue1`	**Noise Level**

Table 22: Event Results in Voice Monitor Short and Continuous Modes (2 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
MOS	The Mean Opinion Score of the waveform analysis. MOS values range from 1 to 5, with 1 standing for poor and 5 for excellent. When the MOS cannot be calculated due to insufficient speech, an invalid MOS value of -999 is reported.	$mos	`@UQMValue2`	**MOS**
Idle Frames	The number of frames before the speech starts. A frame is taken every 0.03 seconds and contains 240 samples as a sample is taken every 0.000125 seconds.	$idleframes	`@UQMValue3`	**Idle Frames**
Tone diagnostic code	The diagnostic information about the tone. The following values apply: • 0: No tones detected. • 1: Modem tones detected. • 2: Fax tones detected. • 4: Undetermined tone detected. • 16: DTMF tones detected. • 32: Pure tones at non-network tone frequencies detected. • 64: Dual tones detected at network tone frequencies. • 128: Pure tones detected at network tone frequencies.	$tonediagnosticcode	`@UQMValue4`	**TDC**
Number of Voice Samples	The number of voice samples (valid frames) used to compute the MOS.	$numberofmos	`@UQMValue5`	No. of Frames
Call setup time	The time taken to set up the call, in seconds.	$setuptime	`@CallSetupTime`	**Call Setup Time**
Call disconnection time	The time taken to disconnect the call (if available). If this measurement is not available, the value N/A is returned. This is a floating point value, measured in seconds.	$ctime	`@CallClosureTime`	**Call closure time**

Table 22: Event Results in Voice Monitor Short and Continuous Modes (3 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Information Code	The information code returned by the Psytechnics NiQA server. This is only set if at least one sample of speech is available, otherwise it contains the default value of 1. The following values apply: • 0: All measurements calculated. The signal was assessed. • 1: Not enough frames to calculate activity or RMS level. Signal was too short to assess. • 2: RMS level outside calibrated range. • 16: Not enough speech was detected to make a valid assessment. • 32: Speech level outside calibrated range. Too much speech was detected to make a valid assessment. • 64: Not enough bi-directional noise frames to make speech level measurement from end A to end B. • 128: Noise level outside calibrated range. • 256: Not enough single-talk speech to detect echo. • 512: The echo did not correlate well enough with the original speech. • 1024: Echo return loss was outside the calibrated range. • 2048: The echo delay was outside the calibrated range. • 8192: Modem or Fax tones were detected, so the signal contained no speech. • 4096: Other tones detected.	$informationcode	`@UQMValue6`	**Voice Code**

Table 22: Event Results in Voice Monitor Short and Continuous Modes (4 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Signal to Noise Ratio (SNR)	SNR measures the speech stream's signal strength relative to background noise. The higher the signal to noise ratio, the better the possible listening quality. SNR is measured in decibels (dB). An SNR value of 20 dB can begin to impair conversational quality.	$snr	`@UQMValue7`	**SNR**
Level Compensated MOS	The level compensated MOS is used to offset for high active speech levels. For example, a speech stream may have high volume but still be of good quality. Values range from 1 to 5, with 1 standing for poor and 5 for excellent.	$lcmos	`@UQMValue8`	**LC MOS**
Speech Activity	The proportion of the call with speech activity. Speech activity is expressed as a percentage (%) of the call that contains speech.	$speechactivity	`@UQMValue9`	**Speech Activity**
Remote party phone number	The telephone number that the mobile handset has called or has been called from. This includes the international and local area code.	$isdn	`@UQMValue10`	**ISDN**
Failure code	The failure code is used to provide a reason for not making a call assessment. The following values apply: • 0: Valid analysis • 1: No voice analysis • 2: Call setup failure • 3: NiQAMobile server failure	$fail	`@UQMValue11`	**Fail**

Table 22: Event Results in Voice Monitor Short and Continuous Modes (5 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Call exit code	The call exit code. If set, it shows why the call terminated. On certain handsets, if the call terminated normally it is set to 0. On other handsets, it set to one of a range of hexadecimal values, which specify why the call terminated (for example, if the remote user hangs up).	$ecode	`@UQMValue12`	**Ecode**
Call direction	The direction of the call. This is a boolean value: • 0: Outgoing call • 1: Incoming call	$dir	`@UQMValue13`	**Direction**
Assessment mode	The voice assessment mode being used. This can hold the values: • Short • Full • Continuous	$assessmentmode	`@UQMValue26`; also part of `@Identifier`	**Assessment Mode**

Measurements in Full Mode

The following table shows which results are reported in events for the Full voice monitoring mode. The table also presents how token values are mapped to the fields, and which AEL columns include the results by default.

Note: Table 23 does not include event data common to all monitors. This is listed in Table 21 on page 92.

Table 23: Event Results in Voice Monitor Full Mode (1 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Speech Level	The active speech level of the speech stream during the call. The speech level should be between -16 dboV and -26 dboV for acceptable speech strength. Values range from 0 to -127 dboV.	$speechlevel	`@UQMValue0`	**Speech Level**
Mean Noise Level	The average noise level of the speech stream. The lower the noise level, the better the listening quality. An acceptable noise level is -40 dboV. Values range from 0 to -127 dboV.	$meannoiselevel	`@UQMValue1`	**Mean Noise Level**
Mean MOS	The average Mean Opinion Score of the waveform analysis. MOS values range from 1 to 5, with 1 standing for poor and 5 for excellent. When the MOS cannot be calculated due to insufficient speech, an invalid MOS value of -999 is reported.	$meanmos	`@UQMValue2`	**Mean MOS**
Idle Frames	The number of frames before the speech starts. A frame is taken every 0.03 seconds and contains 240 samples as a sample is taken every 0.000125 seconds.	$idleframes	`@UQMValue3`	**Idle Frames**

Table 23: Event Results in Voice Monitor Full Mode (2 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Tone Diagnostic Code	The diagnostic information about the tone. The following values apply: • 0: No tones detected. • 1: Modem tones detected. • 2: Fax tones detected. • 4: Undetermined tone detected. • 16: DTMF tones detected. • 32: Pure tones at non-network tone frequencies detected. • 64: Dual tones detected at network tone frequencies. • 128: Pure tones detected at network tone frequencies.	$tonediagnosticcode	`@UQMValue4`	**TDC**
Call duration	The call duration in seconds.	$callduration	`@UQMValue5`	**Call Duration**

Table 23: Event Results in Voice Monitor Full Mode (3 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Information Code	The information code returned by the Psytechnics NiQA server. This is only set if at least one sample of speech is available, otherwise it contains the default value of 1. The following values apply: • 0: All measurements calculated. The signal was assessed. • 1: Not enough frames to calculate activity or RMS level. Signal was too short to assess. • 2: RMS level outside calibrated range. • 16: Not enough speech was detected to make a valid assessment. • 32: Speech level outside calibrated range. Too much speech was detected to make a valid assessment. • 64: Not enough bi-directional noise frames to make speech level measurement from end A to end B. • 128: Noise level outside calibrated range. • 256: Not enough single-talk speech to detect echo. • 512: The echo did not correlate well enough with the original speech. • 1024: Echo return loss was outside the calibrated range. • 2048: The echo delay was outside the calibrated range. • 8192: Modem or Fax tones were detected, so the signal contained no speech. • 4096: Other tones detected.	$informationcode	`@UQMValue6`	**Voice Code**

Table 23: Event Results in Voice Monitor Full Mode (4 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Mean Signal to Noise Ratio	The average Signal to Noise Ratio.	$meansnr	`@UQMVValue7`	**Mean SNR**
Level Compensated MOS	The level compensated MOS is used to offset for high active speech levels. For example, a speech stream may have high volume but still be of good quality. Values range from 1 to 5, with 1 standing for poor and 5 for excellent.	$lcmos	`@UQMVValue8`	**LC MOS**
Speech Activity	The proportion of the call with speech activity. Speech activity is expressed as a percentage (%) of the call that contains speech.	$speechactivity	`@UQMVValue9`	**Speech Activity**
Remote party phone number	The telephone number that the mobile handset has called or has been called from. This includes the international and local area code.	$isdn	`@UQMVValue10`	**ISDN**
Failure code	The failure code is used to provide a reason for not making a call assessment. The following values apply: • 0: Valid analysis • 1: No voice analysis • 2: Call setup failure • 3: NiQAMobile server failure	$fail	`@UQMVValue11`	**Fail**
Call exit code	The call exit code. If set, it shows why the call terminated. On certain handsets, if the call terminated normally it is set to 0. On other handsets, it set to one of a range of hexadecimal values, which specify why the call terminated (for example if the remote user hangs up).	$ecode	`@UQMVValue12`	**Ecode**

Table 23: Event Results in Voice Monitor Full Mode (5 of 5)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Call direction	The direction of the call. This is a boolean value: • 0: Outgoing call • 1: Incoming call	$dir	`@UQMValue13`	**Direction**
Maximum MOS	The maximum Mean Opinion Score of the waveform analysis	$maxmos	`@UQMValue14`	**Max MOS**
Minimum MOS	The minimum Mean Opinion Score of the waveform analysis.	$minmos	`@UQMValue15`	**Min MOS**
Maximum signal to noise ratio	The maximum Signal to Noise Ratio.	$maxsnr	`@UQMValue16`	**Max SNR**
Minimum signal to noise ratio	The minimum Signal to Noise Ratio (SNR). SNR measures the speech stream's signal strength relative to background noise. The higher the signal to noise ratio, the better the possible listening quality. SNR is measured in decibels (dB). An SNR value of 20 dB can begin to impair conversational quality.	$minsnr	`@UQMValue17`	**Min SNR**
Maximum noise level	The maximum noise level of the speech stream.	$maxnoiselevel	`@UQMValue18`	**Max Noise Level**
Minimum noise level	The minimum noise level of the speech stream.	$minnoiselevel	`@UQMValue19`	**Min Noise Level**
Assessment mode	The voice assessment mode being used. This can hold the values: • Short • Full • Continuous	$assessmentmode	`@UQMValue26;` also `@Identifier`	**Assessment Mode**
Call setup time	The time taken to set up the call, in seconds.	$setuptime	`@CallSetupTime`	**Call Setup Time**
Call disconnection time	The time taken to disconnect the call (if available)	$ctime	`@CallClosureTime`	**Call Closure Time**

4.4 Data Monitor Measurements

The results reported by the Data Monitor are also carried by tokens. Many tokens map to the same fields in the ObjectServer rules file and the same columns in the AEL as the Voice Monitor results, although they stand for different measurements.

This section provides details on:

- Core measurements made by the Data Monitor (HTTP analysis mode)
- Measurements made using the Packet Statistics Monitoring feature
- Measurements made using the Interface Monitoring feature

For further information on the Data Monitor and its features, see *About the Data Monitor* on page 23.

Note: The tables in this chapter provide default field mappings and AEL column names. The AEL can be customized using the view builder features of Netcool/Webtop. For details about customizing the column names and views in Netcool/Webtop, see the *Netcool/Webtop Administration Guide*.

Automated Links and Dynamic Content

The Data Monitor is designed to generate a single event for a web page. Any links within the page that are automatically generated, using the Javascript scripting language for example, are not detected by the monitor and are therefore not included in the page event. However, in such cases, the presence of these links may result in extra events being created.

Also, the Data Monitor is not designed to detect Javascript redirected content or other dynamic behaviors. Any web sites using excessive Javascript behavior may not be parsed and monitored successfully, leading to extra events being created.

In general, any other web scripting language that generates web links automatically may contain content that is not included in single web page events generated by the Data Monitor. Such content may result in extra events.

Core Measurements

The following table shows what results are reported in every event for a data monitoring operation in the HTTP analysis mode. The table also presents how token values are mapped to the fields, and which AEL columns include the results by default.

The Data Monitor always sends HTTP Analysis data, regardless of whether the Packet Statistics and Interface Monitoring features are enabled or not.

Note: Table 24 does not include event data common to all monitors. This is listed in Table 21 on page 92.

Table 24: Event Results in Data Monitor HTTP Mode (1 of 4)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Bytes Sent	The number of bytes sent from the handset during a web page query. The total bytes sent is calculated by adding the data from all HTTP requests, including web page and other content.	$bytessent	`@UQMValue1`	**Bytes Sent**
Bytes Received	The number of bytes received during a web page download. The total bytes received is calculated by adding the data from all components on the page.	$bytesreceived	`@UQMValue2`	**Bytes Received**
Data Rate	The speed of the data transmission, based on the response data rate in bits per second (bps). This measurement is the average of all web page component responses.	$datarate	`@UQMValue3`	**Data Rate**
Total Time	The total time, in seconds, of the HTTP transaction from the user query to the web page being fully downloaded. This measurement is the sum of the time taken for each component of the page to download, even if components are downloaded simultaneously.	$totaltime	`@UQMValue4`	**Total Time**
Server IP Address	The IP address of the content server from where the download is requested.	$targetaddress	`@UQMValue5`	**Target Address**
Server IP Port	The content server's port to which the HTTP transmission request was sent.	$targetport	`@UQMValue6`	**Target Port**
Handset's IP Address	The IP address of the mobile handset from which the HTTP transmission request was sent.	$sourceaddress	`@UQMValue7`	**Source Address**
Handset's IP Port	The TCP port of the mobile handset from which the HTTP transmission request was sent.	$sourceport	@UQMValue8; also part of @Identifier	**Source Port**

Table 24: Event Results in Data Monitor HTTP Mode (2 of 4)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Protocol	The protocol the Data Monitor is parsing. This is always set to tcp-http.	$protocol	`@UQMValue9`	**Protocol**
File Name	The name of the monitored URL path or file type transferred in the HTTP transaction.	$filename	`@UQMValue10`;	**File Name**; also included in **Identifier** value.
Host header content	Defines the HTTP host header content. This is a domain name. For example: www.google.com.	$monitorhost	`@UQMValue11`	**Monitor Host**
HTTP protocol version	The version of HTTP protocol being used. For example, 1.1.	$httpversion	`@UQMValue12`	**HTTP Version**
Response Time	The response time, in seconds, between the first packet of a query and the time of the first packet being sent from the server. This is a floating point value.	$responsetime	`@UQMValue13`	**Response Time**
HTTP Parsing Level	The HTTP parsing level of the monitor. This helps identify which stage the data monitor was at when a failure occurred. The following values apply: • 1: Waiting get query • 2: Reading query headers • 3: Waiting server status • 4: Reading server headers • 5: Reading data	$parsinglevel	`@UQMValue14`	**Parsing Level**
HTTP status description	The HTTP status description appearing after the status code. For example, "OK" or "Timeout".	$description	`@UQMValue15`	HTTP Code Description

Table 24: Event Results in Data Monitor HTTP Mode (3 of 4)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
HTTP status	The HTTP status code of the response. This provides information about the result of a request. HTTP status codes are 3 digit codes having the following form: • 1xx: Informational. The request has been received and the process is continuing. • 2xx: Success. The action was successful. • 3xx: Redirection. Further action must be taken in order to complete the request. • 4xx: Client error. The request cannot be completed due to a syntax error. • 5xx: Server error. The server failed to complete the request.	$status	`@UQMValue16`	HTTP Code
Server timeout	The maximum time, in seconds, to wait for a response from the server. A timeout event is sent if, for example, a page has not loaded after the specified timeout. This is an integer value.	$timeout	`@UQMValue17`	Timeout
Number of web page components	The number of components in a single web page. This is an integer value.	$componentnumber	`@UQMValue18`	Page Components Number

Table 24: Event Results in Data Monitor HTTP Mode (4 of 4)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Number of successful component downloads	The number of components with HTTP return status of 200, indicating a successful download. See $status for information on HTTP return codes. This is an integer value.	$component200pages	`@UQMValue19`	Successful Page Component Download
Timed out pages	The number of components that have timed out before being received. This is an integer value.	$componenttmopages	`@UQMValue20`	Timed-out Page Components

Packet Statistics Monitoring

The following table shows what results are reported in every event for a data monitoring operation when using the Packet Statistics Monitoring feature. The table also presents how token values are mapped to the fields, and which AEL columns include the results by default.

Note: Table 25 does not include event data common to all monitors. This is listed in Table 21 on page 92.

Table 25: Event Results returned by Data Monitor Packet Statistics feature (1 of 4)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Minimum jitter	The minimum jitter measured by the data monitor. Jitter is the undesirable variation in the delay of packet delivery. This measurement provides quality of service information. Lower values indicate better quality. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjittermin	`@UQMValue0`	**Minimum Jitter**
Maximum jitter	The maximum jitter measured by the data monitor. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjittermax	`@UQMValue1`	**Maximum Jitter**

Table 25: Event Results returned by Data Monitor Packet Statistics feature (2 of 4)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Mean jitter	The mean jitter measured by the data monitor. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjittermean	`@UQMValue2`	**Mean Jitter**
Sum of jitter	The total jitter, for all packets, measured by the data monitor. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjittersum	`@UQMValue3`	**Sum of Jitter**
Sum squared jitter	The sum squared value for jitter. This is a statistical measurement calculated using $packetjittersum. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjittersumsquare	`@UQMValue4`	**Sum Squared Jitter**
Jitter standard deviation	The jitter standard deviation. This is a statistical measurement calculated using the other jitter measurements. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjitterstddev	`@UQMValue5`	**Std Deviation Jitter**
Packets out of sequence	The number of TCP packets delivered out of sequence. This is an integer value, produced when Packet Statistics is enabled in the data monitor.	$packetsoutofseq	`@UQMValue6`	**Out of Sequence Packets**
Duplicate packets	The number of duplicate packets delivered. This is an integer value, produced when Packet Statistics is enabled in the data monitor.	$packetsduplicates	`@UQMValue7`	**Duplicate Packets**

Table 25: Event Results returned by Data Monitor Packet Statistics feature (2 of 4)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Mean jitter	The mean jitter measured by the data monitor. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjittermean	`@UQMValue2`	**Mean Jitter**
Sum of jitter	The total jitter, for all packets, measured by the data monitor. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjittersum	`@UQMValue3`	**Sum of Jitter**
Sum squared jitter	The sum squared value for jitter. This is a statistical measurement calculated using $packetjittersum. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjittersumsquare	`@UQMValue4`	**Sum Squared Jitter**
Jitter standard deviation	The jitter standard deviation. This is a statistical measurement calculated using the other jitter measurements. This is a floating point value, produced when Packet Statistics is enabled in the data monitor.	$packetjitterstddev	`@UQMValue5`	**Std Deviation Jitter**
Packets out of sequence	The number of TCP packets delivered out of sequence. This is an integer value, produced when Packet Statistics is enabled in the data monitor.	$packetsoutofseq	`@UQMValue6`	**Out of Sequence Packets**
Duplicate packets	The number of duplicate packets delivered. This is an integer value, produced when Packet Statistics is enabled in the data monitor.	$packetsduplicates	`@UQMValue7`	**Duplicate Packets**

Table 25: Event Results returned by Data Monitor Packet Statistics feature (3 of 4)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Total number of packets	The total number of packets passed through the socket connection. This is an integer value, produced when Packet Statistics is enabled in the data monitor.	$packetsnumber	`@UQMValue8`	Total number of Packets
Protocol	The protocol the Data Monitor is parsing. For UQM 2.0, this is tcp-http.	$protocol	`@UQMValue9`	**Protocol**
Maximum length of packets	The payload size, in bytes, of the largest packet transferred. This is an integer value, produced when Packet Statistics is enabled in the data monitor.	$packetsmaxlength	`@UQMValue10`	Packet Max Length
Mean length of packets	The average payload size, in bytes, of packets transferred. The payload is the data inside the packet, not including the header information. This is an integer value, produced when Packet Statistics is enabled in the data monitor.	$packetsmeanlength	`@UQMValue11`	Packet Mean Length
Minimum size of packet payload	The payload size, in bytes, of the smallest packet transferred. This is an integer value, produced when Packet Statistics is enabled in the data monitors.	$packetsminlength	`@UQMValue12`	Packet Min Length
Mobile connection	The IP address and port number of the mobile handset.	$mobileconnection	`@UQMValue13`; also part of `@Identifier`	Mobile Connection
Server connection	The IP address and port number of the server content provider.	$serverconnection	`@UQMValue14`	Server Connection
Assessment mode	Defines whether the event result is based on a stream or a substream. connection. See Video Monitor event results for further information.	$assessmentmode	`@UQMValue26`; also part of `@Identifier`	Assessment Mode
Bytes sent	The number of bytes sent from the handset to the server.	$bytessent	`@UQMValue15`	Bytes Sent

Table 25: Event Results returned by Data Monitor Packet Statistics feature (4 of 4)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Bytes received	The number of bytes received by the handset from the server.	$bytesreceived	`@UQMValue16`	Bytes Received
Total time	The total time, in seconds, between the first packet being sent by the handset and the last packet being received from the server.	$totaltime	`@UQMValue17`	Total Time
Response time	The total time, in seconds, between the first packet being sent by the handset and the first packet being received from the server.	$responsetime	`@UQMValue18`	Response Time
Uplink data rate	The data rate, in bits per second, experienced during transfers from the server to the handset.	$uplinkdatarate	`@UQMValue19`	Uplink Data Rate
Downlink data rate	The data rate, in bits per second, experienced during transfers from the handset to the server.	$downlinkdatarate	`@UQMValue20`	Downlink Data Rate

Interface Monitoring

The following table shows what results are reported in every event for a data monitoring operation when using the Interface Monitoring feature. The table also presents how token values are mapped to the fields, and which AEL columns include the results by default.

Note: Table 26 does not include event data common to all monitors. This is listed in Table 21 on page 92.

Table 26: Event Results returned by Data Monitor Interface Monitoring Feature (1 of 2)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Interface name	The interface name of the connection. This can refer to both failed and successful connections.	$interfacename	`@UQMValue0`	**Interface Name**
Interface identity	The interface identity of the connection. This can refer to both failed and successful connections.	$interfaceiapid	`@UQMValue1`	**Interface Identity**

Table 26: Event Results returned by Data Monitor Interface Monitoring Feature (2 of 2)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Interface error	The interface error. This defines the error that has caused a failed connection.	$interfaceerror	`@UQMValue2`	**Interface Error**
Connection time	The time taken to establish a connection, in seconds.	$connectiontime	@UQMValue3	**Connection Time**

4.5 Video Monitor Measurements

The results reported by the Video Monitor are also carried by tokens. Many tokens map to the same fields in the ObjectServer rules file and the same columns in the AEL as the Voice Monitor results, although they stand for different measurements.

Many of the video monitor measurements provide information on entire streams or parts of streams called substreams. The value of `$type` indicates whether the event result is based on a stream or a substream.

Note: Most video monitor measurements can be used for both streams and substreams. Where this is not the case, this is stated in the token description in Table 27.

The following table shows which results are reported in every event for a video monitoring operation. The table also shows how token values are mapped to the fields, and which AEL columns include the results by default.

Note: Table 27 does not include event data common to all monitors. This is listed in Table 21 on page 92.

Table 27: Event Results for Video Monitor (1 of 8)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Stream duration	The duration of the video stream in ms.	$duration	`@UQMValue0`	**Duration**
Average number of consecutive packets lost	If there are multiple instances of consecutive packet loss in a single video stream, this is the average number of consecutive packets lost. This is an integer value.	$aveconsecpl	`@UQMValue1`	**Average Packets Lost**
Maximum number of consecutive packets lost	The maximum number of consecutive packets lost in the duration of a video stream. This measurement provides information about the severity of the longest gap in video playback. This is related to $gaplength. This is an integer value.	$maxconsecpl	`@UQMValue2`	**Max Packets Lost**

Table 27: Event Results for Video Monitor (2 of 8)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Mean absolute value for jitter	The mean absolute value of jitter measured by the video monitor. Jitter is the undesirable variation in the delay of packet delivery. This measurement provides quality of service information. Lower values indicate better quality. This is a floating point value.	$ave1ptabspdvms	`@UQMValue3`	**Average Packets Lost**
Expected number of packets	The number of packets expected to be received in the duration of the video stream. This is an integer value.	$exppkts	`@UQMValue4`	**Expected Packets**
Maximum value of jitter	The maximum value of jitter experienced during a stream. This measurement provides quality of service information. Lower values indicate better quality. This is a floating point value.	$maxposjitter	`@UQMValue5`	**Max. Pos. Jitter**
Minimum value of jitter	The minimum value of jitter experienced during a stream. This measurement provides quality of service information. Lower values indicate better quality. This is a floating point value.	$minnegjitter	`@UQMValue6`	**Min. Neg. Jitter**
Number of missed packets	The number of packets missed due to them being out of sequence. This measurement provides quality of service information. Lower values indicate better quality. This is an integer value.	$missedpkts	`@UQMValue7`	**Missed Packets**
Transport protocol	The transport protocol to be monitored. This has a fixed value of `rtp/rtcp/rtsp`	$protocol	`@UQMValue9`	**Protocol**

Table 27: Event Results for Video Monitor (3 of 8)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Mean burst length	The mean duration of all burst periods that have occurred since the beginning of a video stream. A burst period is the constant delivery of video without any gaps. Burst length is expressed in ms. If no actual values are available, estimates must be used. Also, if there have been no burst periods, then $burstlength = 0.	$burstlength	`@UQMValue10`	**Burst Length**
Burst density	The proportion of the video stream that has been free from gaps in playback. This is expressed as a percentage.	$burstdensity	`@UQMValue11`	**Burst Density**
Mean gap length	The mean duration of all gaps that have occurred since the beginning of a video stream. Gap length is expressed in ms. If no actual values are available, estimates must be used.	$gaplength	`@UQMValue12`	**Gap Length**
Gap density	The proportion of the video stream that has consisted of gaps in playback. This is expressed as a percentage.	$gapdensity	`@UQMValue13`	**Gap Density**
Mean opinion score of video stream	A video quality measurement score, for a given sample period, based on the ITU Mean Opinion Score (MOS) scale. The scale represents customer perceptions of quality. MOS values range from 1 to 5, with 1 standing for poor and 5 for excellent.	$mos	`@UQMValue14`	**MOS**
Number of packets received for diagnostics	The number of packets in the stream that relate to diagnostics. This is an integer value.	$rcvpkts	`@UQMValue15`	**Received Packets**

Table 27: Event Results for Video Monitor (4 of 8)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Number of lost packets	Packet loss is common when using the UDP protocol. This is the number of packets that have been lost in the duration of a stream. This is an integer value.	$lostpkts	`@UQMValue16`	**Lost Packets**
Quality degradation	A real-time measure of the current network quality, relative to a theoretical perfect quality network. This is a floating point value. Higher values represent lower quality, with 0 corresponding to perfect quality.	$netmosdeg	`@UQMValue17`	**MOS Degradation**
Quality degradation due to video compression	The proportion of the quality degradation that is due only to video compression. Used with $propduejitter and $propduepktloss, this can be used to analyze reasons for quality degradation. This measure is expressed as a percentage.	$propduecompress	`@UQMValue18`	**Compress Degradation**
Quality degradation due to jitter	The proportion of the quality degradation that is due only to variations in packet delay. Used with $propduecompress and $propduepktloss, this can be used to analyze reasons for quality degradation. This measure is expressed as a percentage.	$propduejitter	`@UQMValue19`	**Jitter Degradation**

Table 27: Event Results for Video Monitor (5 of 8)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Quality degradation due to packet loss	The proportion of the quality degradation that is due only to packet loss. Used with $propduecompress and $propduejitter, this can be used to analyze reasons for quality degradation. This measure is expressed as a percentage.	$propduepktloss	`@UQMValue20`	**Pkt. Loss Degradation**
Total packets in stream	The total number of packets in the stream. $totalpackets cannot be used for substreams.	$totalpackets	`@UQMValue21`	**Total Packets**
Codec identifier	Identifies the video codec being used. $codectype cannot be used for substreams.	$codectype	`@UQMValue22`	**Codec Type**
Number of video frames	The number of video frames delivered in the stream. $framenb cannot be used for substreams.	$framenb	`@UQMValue23`	**Num. Video Frames**
Number of octets transmitted	The number of octets transmitted in the stream. An octet is a group of 8 bits, so the number of octets is the same as the number of bytes. $datasz cannot be used for substreams.	$datasz	`@UQMValue24`	**Data Size**
Number of RTP packets	The number of RTP packets in the stream. See $protocol for further information. $numpkts cannot be used for substreams.	$numpkts	`@UQMValue25`	**Num. Packets**

Table 27: Event Results for Video Monitor (6 of 8)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Assessment mode	Defines whether the event result is based on a stream or a substream. When the video monitor detects a video stream, it sends periodic results to the Control Function. These periodic results are based on a section of video, or a substream. When the video has stopped streaming, the monitor sends results based on the entire stream. You can configure the sampling interval of an event. This lets you define whether the event is based on a stream or a substream.	$assessmentmode	`@UQMValue26`; also part of `@Identifier`	**Assessment Mode**
E-Model score	The E-Model is a tool for providing an estimate of the user's satisfaction with network quality. At present, video analysis is performed, but not sound on video analysis. For this reason, `emodelie` has a fixed value of 0.	$emodelie	`@UQMValue36`	**Emodelie**
Congestion due to slow access to packets	The amount of degradation in network quality due to packet congestion. This is expressed as a percentage.	$paccesscongestion	`@UQMValue37`	**Access Congestion / %**
Packets being delivered out of sequence	The amount of degradation in network quality due to packets being delivered out of sequence. This is expressed as a percentage.	$pdiverserouting	`@UQMValue38`	**Divert Routing / %**
Congestion due to end devices	The amount of degradation in network quality due to end devices such as hubs and switches. This is expressed as a percentage.	$plancongestion	`@UQMValue39`	**LAN Congestion / %**
Congestion due to routers	The amount of degradation in network quality due to router processing problems. This is expressed as a percentage.	$proutercongestion	`@UQMValue40`	**Router Congestion / %**

Table 27: Event Results for Video Monitor (7 of 8)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Unreliable link	The amount of degradation in network quality due to unreliable network links. This is expressed as a percentage.	$punreliablelink	`@UQMValue41`	**Bad Link / %**
Estimated delay	The estimated delay for incoming packets. $delayest cannot be used for substreams.	$delayest	`@UQMValue42`	**Delay Estimate**
Packets disregarded	The amount of packets estimated to be discarded due to delay (jitter). This is expressed as a percentage. $discardest cannot be used for substreams.	$discardest	`@UQMValue43`	**Discard Estimate**
Network Source Address	The network source address. $nwsrc cannot be used for substreams.	$nwsrc	`@UQMValue44`	**Network Src. Addr.**
Network Destination Address	The network destination address. $nwdst cannot be used for substreams.	$nwdst	`@UQMValue45`	**Network Dst. Addr.**
Transport Source Address	The transport source address. $tpsrc cannot be used for substreams.	$tpsrc	`@UQMValue46`	**Transport Src. Addr.**
Transport Destination Address	The transport destination address. $tpdst cannot be used for substreams.	$tpdst	`@UQMValue47`	**Transport Dst. Addr.**

Table 27: Event Results for Video Monitor (8 of 8)

Information	Description and Usage	Token Carrying Result	Field in Rules File	Default Column(s) in AEL
Capture time of first packet	The time that the first packet in the stream was captured. This is quoted in UTC format, as number of seconds elapsed since 01/01/1970 00:00:00. For example, 1131179339. $starttimesec cannot be used for substreams.	$starttimesec	`@UQMValue48`	**Start Time / sec**
Capture time of latest packet	The time that the latest packet in the stream was captured. $latesttmese cannot be used for substreams.	$latesttimesec	`@UQMValue49`	**Latest Time / sec**

Appendix A: XML File Example

This appendix provides a full example of a valid XML configuration file for the Netcool for Wireless User Quality Monitoring unit.

This appendix contains the following section:

A.1 XML Configuration File

The following is an XML configuration file with example settings.

Example

```
[1]  <cfengine compatible='1' release='1'>
[2]    <configfetcher>
[3]      <fetchperiod>43200</fetchperiod>
[4]      <maxperiodextensionpercent>10</maxperiodextensionpercent>
[5]      <url>http://uqm.corp-2.com/cfg/dev/demo_config.xml</url>
[6]    </configfetcher>
[7]    <producer compatible='1' release='1'>
[8]      <waitfornet checkintervalseconds='20' retries='2'>
[9]        <net country='234' network='15'>accept</net>
[10]     </waitfornet>
[11]     <monitor type='voicemonitor' sysid=''271000008' compatible='1' release='1'>
[12]       <numsamples>5</numsamples>
[13]       <samplinginterval>6</samplinginterval>
[14]       <heartbeat>5</heartbeat>
[15]       <mode>continuous</mode>
[16]     </monitor>
[17]     <monitor type='videomonitor' sysid='536877773' compatible='1' release='1'>
[18]       <sampleperiod>30</sampleperiod>
[19]       <config label='all'>
[20]         <ipaddress>0.0.0.0</ipaddress>
[21]         <netmask>0.0.0.0</netmask>
[22]       </config>
[23]     </monitor>
[24]     <monitor type='datamonitor' sysid='271000010' compatible='1' release='1'>
[25]       <timeout>120</timeout>
[26]       <packetstats enable='true' protocol='any'>
[27]         <poll enable='false' />
[28]         <ipfilter>
[29]           <ipaddress>0.0.0.0</ipaddress>
[30]           <netmask>0.0.0.0</netmask>
[31]         </ipfilter>
[32]       </packetstats>
[33]       <ifmonitor enable='true'>
[34]         <apnname>ProviderX live!</apnname>
[35]         <apnname>internet</apnname>
[36]         <apnname>K2</apnname>
[37]       </ifmonitor>
[38]       <config label='all'>
[39]         <ipaddress>0.0.0.0</ipaddress>
[40]         <netmask>0.0.0.0</netmask>
[41]         <rootpage>*</rootpage>
[42]         <filename>*</filename>
[43]         <hostname>*</hostname>
[44]       </config>
```

```
[45]      </monitor>
[46]      <monitor type='activeclient' sysid='536877768' compatible='1' release='1'>
[47]        <splashscreen>on</splashscreen>
[48]        <periodicity>120</periodicity>
[49]        <schedule>
[50]          <action type='delay'>
[51]            <interval>30</interval>
[52]          </action>
[53]          <action type='call'>
[54]            <phone>121</phone>
[55]          </action>
[56]          <action type='delay'>
[57]            <interval>35</interval>
[58]          </action>
[59]          <action type='hangup'/>
[60]          <action type='delay'>
[61]            <interval>60</interval>
[62]          </action>
[63]          <action type='stream'>
[64]           <uri>rtsp://stream.corp-2.com/IT/200206_030_00_IT.3gp?plan</uri>
[65]            <apn>internet</apn>
[66]            <volume>0</volume>
[67]            <duration>60</duration>
[68]          </action>
[69]          <action type='httpget'>
[70]            <uri>www.content.com/index.html</uri>
[71]            <apn>AnAPN</apn>
[72]          </action>
[73]        </schedule>
[74]      </monitor>
[75]      <queue>
[76]        <size>10</size>
[77]      </queue>
[78]      <setresults>
[79]        <set name='customername'>Corp-2</set>
[80]      </setresults>
[81]      <filter>
[82]        <action>
[83]          <eval>
[84]            <lefthand>token:servicetype</lefthand>
[85]            <operation>equal</operation>
[86]              <case>
[87]                <value>VOICE</value>
[88]                <action>
[89]                  <eval>
[90]                    <lefthand>token:mos</lefthand>
[91]                    <operation>lte</operation>
[92]                    <case>
[93]                      <value>3.5</value>
[94]                      <action><accept/></action>
[95]                    </case>
```

```
[96]                    <default><action><reject/></action></default>
[97]                  </eval>
[98]                </action>
[99]              </case>
[100]             <default>
[101]               <action><accept/></action>
[102]             </default>
[103]         </eval>
[104]       </action>
[105]     </filter>
[106]     <positionfinder>
[107]       <btport>1</btport>
[108]       <btname>GPS-31829</btname>
[109]     </positionfinder>
[110]     <maxageseconds>3600</maxageseconds>
[111]   </producer>
[112]   <iap source='commprefselect'>
[113]     <preconditions>exclusive3g</preconditions>
[114]     <commprefselector compatible='1' release='1'>
[115]       <operator type='configfetcher' country='234' networkcode='15'>
[116]         <apn>
[117]           <name>internet</name>
[118]           <username>web</username>
[119]           <password>web</password>
[120]         </apn>
[121]       </operator>
[122]       <operator country='234' network='30'>
[123]         <apn>
[124]           <name>general.mobile-provider.uk</name>
[125]           <username>user</username>
[126]           <password>mobile</password>
[127]         </apn>
[128]       </operator>
[129]       <operator country='234' network='31'>
[130]         <apn>
[131]           <name>general.mobile-provider.uk</name>
[132]           <username>user</username>
[133]           <password>mobile</password>
[134]         </apn>
[135]       </operator>
[136]       <operator country='234' network='32'>
[137]         <apn>
[138]           <name>general.mobile-provider.uk</name>
[139]           <username>user</username>
[140]           <password>mobile</password>
[141]         </apn>
[142]       </operator>
[143]     </commprefselector>
[144]   </iap>
[145]   <reporter compatible='1' release='1'>
[146]     <iptransmitter>
```

```
[147]       <blackoutfactor>5</blackoutfactor>
[148]       <blackoutmin>1</blackoutmin>
[149]       <blackoutmax>60</blackoutmax>
[150]       <backofffactor>1.5</backofffactor>
[151]       <backoffmin>3</backoffmin>
[152]       <backoffmax>3600</backoffmax>
[153]     </iptransmitter>
[154]     <reports>
[155]       <ipreport type='udp'>
[156]         <ipaddress>62.190.48.178</ipaddress>
[157]         <port>9520</port>
[158]         <maxencrypted>1200</maxencrypted>
[159]         <maxconcatenated>4096</maxconcatenated>
[160]       </ipreport>
[161]     </reports>
[162]   </reporter>
[163] </cfengine>
```

Index

Contact Information

Corporate

Region	Address	Telephone	Fax	World Wide Web
USA	Micromuse Inc. (HQ) 650 Townsend Street San Francisco CA 94103 USA	1-800-Netcool (638 2665) +1 415 568 9800	+1 415 568 9801	http://www.micromuse.com
Europe	Micromuse Ltd. Disraeli House 90 Putney Bridge Road London SW18 1DA United Kingdom	+44 (0) 20 8875 9500	+44 (0) 20 8875 9995	http://www.micromuse.co.uk
Asia-Pacific	Micromuse Ltd. Level 25 77 St Georges Terrace Perth WA 6000 Australia	+61 (0) 8 9213 3400	+61 (0) 8 9325 5030	http://www.micromuse.com.au

Technical Support

Region	Telephone	Fax
USA	1-800-Netcool (800 638 2665) +1 415 568 9800 (San Francisco)	+1 415 568 9801
Europe	+44 (0) 20 8877 0073 (London, UK)	+44 (0) 20 8875 0991
Asia-Pacific	+61 (0) 8 9213 3470 (Perth, Australia) +10 800 852 1012 (North China) +10 800 152 1012 (South China)	+61 (0) 8 9486 1116

Online

Team	E-Mail	World Wide Web
Licensing	Temporary Licenses: temp.licensing@micromuse.com Permanent Licenses: perm.licensing@micromuse.com	support.micromuse.com/helpdesk/licenses
Support	support@micromuse.com	support.micromuse.com