

Oracle Utilities Energy Information Platform

Configuration Guide, Volume 2

Release 1.6.1.0 for Windows

E18200-01

July 2010

Oracle Utilities Energy Information Platform/Energy Information Platform Configuration Guide, Volume 2,
Release 1.6.1.0 for Windows

E18200-01

Copyright © 1999, 2010 Oracle and/or its affiliates. All rights reserved.

Primary Author: Lou Proseri

Contributing Author: Adam Steiner

Contributor: Steve Pratt

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle America, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

This software or hardware and documentation may provide access to or information on content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.

NOTIFICATION OF THIRD-PARTY LICENSES

Oracle Utilities software contains third party, open source components as identified below. Third-party license terms and other third-party required notices are provided below.

License: Apache 1.1

Module: Crimson v1.1.1, Xalan J2

Copyright © 1999-2000 The Apache Software Foundation. All rights reserved.

Use of Crimson 1.1.1 and Xalan J2 within the product is governed by the following (Apache 1.1):

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution. (3) The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>). " Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear. (4) Neither the component name nor Apache

Software Foundation may be used to endorse or promote products derived from the software without specific prior written permission. (5) Products derived from the software may not be called “Apache”, nor may “Apache” appear in their name, without prior written permission.

THIS SOFTWARE IS PROVIDED “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.

License: CoolServlets.com

Module: CS CodeViewer v1.0 (Sun JRE Component)

Copyright © 1999 by CoolServlets.com

Use of this module within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution. (3) Neither the component name nor CoolServlets.com may be used to endorse or promote products derived from the software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY COOLSERVLETS.COM 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 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."

License: Justin Frankel, justin@nullsoft.com

Module: NSIS 1.0j (Sun JRE Component)

Use of this module within the product is governed by the following:

(1) The origin of the module must not be misrepresented, and Oracle may not claim that it wrote the original software. If Oracle uses this module in a product, an acknowledgment in the product documentation is appreciated but not required. (2) Altered source versions of the module must be plainly marked as such, and must not be misrepresented as being the original software. (3) The following notice may not be removed or altered from any source distribution: “Justin Frankel justin@nullsoft.com”.

License: ICU4j License

Module: ICU4j

Copyright © 1995-2001 International Business Machines Corporation and others. All rights reserved.

Oracle may use the software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, and/or sell copies of the software, and to permit persons to whom the software is furnished to do so, provided that the above copyright notice and the permission notice appear in all copies of the software and that both the above copyright notice and the permission notice appear in supporting documentation.

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 OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE BE LIABLE FOR ANY CLAIM, OR 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.

License: Info-ZIP

Module: INFO-ZIP ZIP32.DLL (Binary Form)

Copyright (c) 1990-2005 Info-ZIP. All rights reserved

Use of this dll within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the definition and disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the definition and disclaimer below in the documentation and/or other materials provided with the distribution. The sole exception to this condition is redistribution of a standard UnZipSFX binary (including SFXWiz) as part of a self-extracting archive; that is permitted without inclusion of this license, as long as the normal SFX banner has not been removed from the binary or disabled. (3) Altered versions--including, but not limited to, ports to new operating systems, existing ports with new graphical interfaces, and dynamic, shared, or static library versions--must be plainly marked as such and must not be misrepresented as being the original source. Such altered versions also must not be misrepresented as being Info-ZIP releases--including, but not limited to, labeling of the altered versions with the names "Info-ZIP" (or any variation thereof, including, but not limited to, different capitalizations), "Pocket UnZip," "WiZ" or "MacZip" without the explicit permission of Info-ZIP. Such altered versions are further prohibited from misrepresentative use of the Zip-Bugs or Info-ZIP e-mail addresses or of the Info-ZIP URL(s). (4) Info-ZIP retains the right to use the names "Info-ZIP," "Zip," "UnZip," "UnZipSFX," "WiZ," "Pocket UnZip," "Pocket Zip," and "MacZip" for its own source and binary releases.

[Definition]: For the purposes of this copyright and license, "Info-ZIP" is defined as the following set of individuals:

Mark Adler, John Bush, Karl Davis, Harald Denker, Jean-Michel Dubois, Jean-loup Gailly, Hunter Goatley, Ed Gordon, Ian Gorman, Chris Herborth, Dirk Haase, Greg Hartwig, Robert Heath, Jonathan Hudson, Paul Kienitz, David Kirschbaum, Johnny Lee, Onno van der Linden, Igor Mandrichenko, Steve P. Miller, Sergio Monesi, Keith Owens, George Petrov, Greg Roelofs, Kai Uwe Rommel, Steve Salisbury, Dave Smith, Steven M. Schweda, Christian Spieler, Cosmin Truta, Antoine Verheijen, Paul von Behren, Rich Wales, Mike White

[Disclaimer:] "This software is provided "as is," without warranty of any kind, express or implied. In no event shall Info-ZIP or its contributors be held liable for any direct, indirect, incidental, special or consequential damages arising out of the use of or inability to use this software."

License: Paul Johnston

Modules: md5.js

Copyright (C) Paul Johnston 1999 - 2002

Use of these modules within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution. (3) Neither the component name nor the names of the copyright holders and contributors may be used to endorse or promote products derived from the 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.

License: Jef Poskanzer

Modules: DES, 3xDES (Sun JRE Components)

Copyright © 2000 by Jef Poskanzer <jef@acme.com>. All rights reserved

Use of these modules within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution. (3) Neither the component name nor the name of Jef Poskanzer may be used to endorse or promote products derived from the software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR 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 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.

License: Sun Microsystems, Inc.

Modules: Sun Swing Tutorials

Copyright© 1995-2006 Sun Microsystems, Inc. All Rights Reserved.

Use of these modules within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution. (3) Neither the component name nor the name of Sun Microsystems, Inc. and contributors may be used to endorse or promote products derived from the software without specific prior written permission. (4) Oracle must acknowledge that the software is not designed, licensed or intended for use in the design, construction, operation or maintenance of any nuclear facility.

THIS SOFTWARE IS PROVIDED "AS IS," WITHOUT A WARRANTY OF ANY KIND. ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS AND WARRANTIES, INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT, ARE HEREBY EXCLUDED. SUN MICROSYSTEMS, INC. ("SUN") AND ITS LICENSORS SHALL NOT BE LIABLE FOR ANY DAMAGES SUFFERED BY LICENSEE AS A RESULT OF USING, MODIFYING OR DISTRIBUTING THIS SOFTWARE OR ITS DERIVATIVES. IN NO EVENT WILL SUN OR ITS LICENSORS BE LIABLE FOR ANY LOST REVENUE, PROFIT OR DATA, OR FOR DIRECT, INDIRECT, SPECIAL, CONSEQUENTIAL, INCIDENTAL OR PUNITIVE DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF THE USE OF OR INABILITY TO USE THIS SOFTWARE, EVEN IF SUN HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

License: Tom Wu

Module: jsbn library

Copyright © 2003-2005 Tom Wu. All rights reserved

Use of this module within the product is governed by the following:

(1) Redistributions of source code must retain the above copyright notice, this list of conditions and the disclaimer below. (2) Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the disclaimer below in the documentation and/or other materials provided with the distribution.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL TOM WU BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES

OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.

Contents

What's New

New Features in the Oracle Utilities Energy Information Platform Configuration Guide	1-i
New Features for Release 1.6.0.0	1-i
New Features for Release 1.6.1.0	1-ii

Chapter 1

Overview	1-1
Configuration Overview	1-2
What is this book?	1-3
International and Multiple Currency Support	1-5
Locales and Languages	1-5
Date and Numeric Presentation	1-6
Currency Presentation	1-6
Multiple Currency Support	1-8
File Formats	1-9
Database Support	1-9
Notes, Cautions, and Warnings	1-10

Part One

Configuration	1-i
---------------------	-----

Chapter 2

Configuration Files	2-1
LODESTAR.CFG	2-2
Allowable Settings	2-3
EXTENDEDSTATUSES.CFG.XML Configuration File	2-12
IDV.CFG.XML	2-15
INTDCONFIG.CFG.XML	2-17
LOCALES.CFG.XML	2-22
LSCALENDAR.CFG.XML	2-23
LSDB.CFG.XML	2-26
LSLOGGER.CFG.XML	2-27
LSRELAY.CFG.XML	2-34
LSREPORTMONITOR.CFG.XML	2-35
LSSCHDLR.CFG.XML	2-40
LSSECURE.CFG.XML	2-41
LTMH.CFG.XML	2-45
Other Configuration Files	2-51
Tools Files (*.TLS)	2-51
Encrypting Configuration Files	2-53

Chapter 3

Setting Up Energy Information Platform Database Records	3-1
Setting Up Energy Information Platform Lookup Tables.....	3-2
Account Statuses	3-2
Bill Determinants	3-2
End Use.....	3-3
Industrial Classifications	3-3
Jurisdiction	3-3
LSCurrency	3-3
Operating Company	3-4
Revenue Code.....	3-4
Second Per Interval.....	3-4
Standard Industrial Classification (SIC) Code	3-5
Unit of Measure (UOM)	3-5
Overriding and/or Extending the Default UOM List	3-6
Setting Up Other Records.....	3-7
Address	3-8
Address Format.....	3-8
Business Calendar	3-9
Business Day.....	3-9
CIS Account.....	3-9
Contact Method Type	3-9
Contact Purpose.....	3-10
Directory.....	3-10
Factor	3-11
Factor Values	3-11
Group Category.....	3-12
Interrogations	3-12
Interval Data Sources	3-12
LS Address	3-13
LS Market	3-13
LS Premise	3-14
Market Attribute.....	3-14
Market Attribute Type.....	3-14
Market Participant Relationship Type	3-14
Market Participant Type.....	3-15
MV90 Status Code Map.....	3-15
Organization	3-15
Override.....	3-15
Participant Type	3-16
Web Service	3-16
Web Service Type	3-16

Chapter 4

Configuring Interval Data Viewer and Interval Data Manager	4-1
Configuring Interval Data Viewer	4-2
Prepare Interval Data	4-2
Set Up Interval Data Sources.....	4-2
Assign Data and Feature Privileges.....	4-2
Configuring Interval Data Manager.....	4-3
Prepare Interval Data	4-3
Set Up Interval Data Sources.....	4-4
Assign Data and Feature Privileges.....	4-4
Set Up Validation Environment File (Optional).....	4-4

Chapter 5

Configuring Work Queues	5-1
Work Queues Tables Overview	5-2
Work Queues Actions	5-3
Opening a New Item.....	5-3
Listing Items	5-3
Assigning an Item	5-4
Unassigning an Item	5-4
Updating an Item	5-5
Resolving an Item	5-5
Approving an Item	5-5
Rejecting an Item	5-6
Closing an Item	5-7
Reopening an Item.....	5-7
Defining Your Work Queues Configuration	5-8
Defining Work Queues	5-8
Defining Work Queue Types	5-8
Configuring Work Queues and Work Queue Types	5-11
Setting Up Lookup Tables.....	5-11
Setting Up Work Queue Types.....	5-12
Creating Work Queue Items.....	5-20
Using Oracle Utilities Billing Component	5-20
Using the Oracle Utilities Rules Language.....	5-22
Using External Applications.....	5-24

Chapter 6

Setting Up, Configuring, and Running the Energy Information Platform Adapter	6-1
Defining the Adapter Implementation.....	6-2
Defining Business Requirements.....	6-2
Defining and Naming Adapter Servers	6-6
Installing the Adapter Software.....	6-7
Install the Adapter Software.....	6-7
Set Up the LTMH.CFG.XML File.....	6-7
Install JMS Servers and Queues (Optional)	6-8
Configuring the Adapter	6-9
Configuring Adapter Components.....	6-9
Configuring Business Rules	6-11
Configuring Adapter Services	6-17
Using Adapter Data Converters.....	6-22
CSV Parsers	6-22
XML Converters	6-31
Interval Data Converters	6-32
Payload Splitter.....	6-34
ZIP Converters.....	6-38
Using Data Converters with Adapter Services	6-39
Creating Custom Converters.....	6-39
Implementing Adapter Services and Business Rules	6-43
Implementing Import Services and Business Rules	6-43
Implementing Import Interval Services and Business Rules.....	6-45
Implementing Interval Services and Business Rules	6-47
Implementing COM Services and Business Rules.....	6-48
Implementing Assembly Services and Business Rules	6-51
Implementing CustomQuery Services and Business Rules	6-57
Implementing Rate Services and Business Rules	6-59
Implementing Oracle Utilities Meter Data Management Services and Business Rules	6-61

Business Rule Examples.....	6-62
Import Business Rule Example	6-62
Import Interval Business Rule Example	6-65
Interval Business Rule Example	6-68
COM Business Rule Example.....	6-70
Rate Business Rule Example.....	6-71
Assembly Business Rule Example.....	6-72
CustomQuery Business Rule Example.....	6-74
The Adapter Server	6-76
Set Up the LTMH.CFG.XML File.....	6-76
Starting the Adapter Server	6-76
The Adapter Monitor.....	6-77
Starting the Adapter Monitor.....	6-77
Runtime Services Tab.....	6-77
Runtime Queue Tab	6-78
Messaging Tool Tab	6-78
Securing and Encrypting the Adapter Server and Monitor	6-79
Technical Overview	6-79
Creating Certificates.....	6-79
Enabling Secure Communications	6-81

Chapter 7

Configuring Energy Information Platform Security	7-1
Security Features.....	7-2
Framework Features	7-3
User Preferences Features.....	7-4
Set Preferences Features	7-4
Options.....	7-4
Entity Management Features.....	7-5
Common Features	7-8
Securing Interval Data Sources	7-10
Work Queue Features.....	7-11
Work Queues Features.....	7-11
Securing Work Queue Items	7-12
Adapter Features	7-13
Data Source Features	7-14
Securing Reports.....	7-15
Overview	7-15
Securing the Contents of the Run Reports Screen	7-15
Securing the Contents of the View Reports Screen	7-16

Chapter 8

Setting Up Processing to run in Batch Mode.....	8-1
Batch Executables Overview.....	8-2
Data Utilities	8-2
Billing.....	8-3
Miscellaneous.....	8-3
Importing Data	8-4
Importing Customer Data	8-4
Importing Interval Data.....	8-9
Importing Enhanced Interval Data.....	8-26
Exporting Data	8-29
Exporting and Deleting Customer Data	8-29
Exporting and Deleting Interval Data.....	8-35
Executing Oracle Utilities Rules Language Rate Schedules.....	8-41
Purpose of RUNRS	8-41

RUNRS Inputs	8-41
RUNRS Command Syntax	8-42
Automated Profiling	8-48
Purpose of AUTOPROF.....	8-48
AUTOPROF Inputs and Other Prerequisites	8-48
AUTOPROF Command Syntax.....	8-50
AUTOPROF Processing.....	8-52
Migrating RCDATA.....	8-53
Purpose of RCTORDBM.....	8-53
RCTORDBM Command Syntax.....	8-54
 Chapter 9	
Creating a CIS Transaction Record Output File.....	9-1
Transaction Type Description File Format.....	9-2
Output File.....	9-2
Control File	9-3
Sample CIS Transaction File	9-8
Print Detail.....	9-9
 Chapter 10	
Customizing the Energy Information Platform	10-1
Overview.....	10-2
What Sort of Customization Is Possible?.....	10-2
Basic File and Directory Structure	10-2
Customization Ground Rules	10-3
A Note About Dictionary Files	10-3
Creating Custom Menus and Screens.....	10-4
Adding Custom Menu Items.....	10-4
Securing Custom Menu Items.....	10-7
Calling Existing Screens.....	10-13
Creating Custom Pages	10-20
 Part Two	
System Administration.....	1-i
 Chapter 11	
Oracle Utilities Energy Information Platform Services	11-1
Interval Data Storage and Access	11-2
Standard Interval Data Storage.....	11-2
Enhanced Interval Data Storage.....	11-12
Interval Data Access.....	11-21
Interval Data Versioning.....	11-21
Energy Information Platform Report Monitor	11-23
Energy Information Platform Report Scheduler.....	11-24
Energy Information Platform Logging Framework	11-25
Logging Framework Output File Format	11-25
Default Behavior	11-26
Merging Tracing and Logging Files.....	11-28
Tracing Levels.....	11-29
Database Auditing.....	11-36
Types of Database Auditing.....	11-36
Storing Auditing Information	11-36
Installing Auditing.....	11-37
Auditing Configuration and Administration.....	11-39
Auditing Reports	11-39

Chapter 12

Energy Information Platform Security	12-1
Overview	12-2
How Security Works.....	12-2
What Applications Are Secured?	12-2
Security Concepts and Data.....	12-3
Security Modes	12-3
Types of Security.....	12-3
Security Data.....	12-4
Working with Function-Level Security	12-7
Types of Function-Level Security	12-7
Applying Function-Level Security	12-7
Working with Record-Level Security	12-9
Types of Record-Level Security.....	12-9
Applying Record-Level Security	12-12
Examples of Using Record-Level Security.....	12-12
Limits on Record-level Security.....	12-13
The Security User Interface	12-14
Setting Up Security.....	12-15
Setting Up Security in Default Mode.....	12-15
Setting Up Security in LDAP Mode.....	12-20
Integrating Security with LDAP.....	12-23
Overview	12-23
Integration Details	12-23
Troubleshooting.....	12-26
Useful Tools.....	12-26
Connecting to LDAP Servers over SSL	12-27
Single Sign-On	12-29
LSSECURE.CFG.XML File	12-29
Creating Energy Information Platform User IDs for Single Sign-On	12-30
IIS Configuration	12-30

Chapter 13

Energy Information Platform Reporting Framework	13-1
Overview	13-2
Reporting Framework Components	13-2
How the Energy Information Platform Reporting Framework Works	13-4
Supported Report Formats.....	13-5
Energy Information Platform Reporting Framework Database Tables	13-6
Report Category	13-6
Report Instance	13-6
Report Priority.....	13-7
Report Template	13-7
Report Template Product.....	13-8
Report Type	13-8
Report Unit	13-9
The Report Administration User Interface	13-10
Running and Viewing Reports	13-11
Running Reports from the User Interface	13-11
Running Reports from the Oracle Utilities Rules Language.....	13-11
Running Reports in Batch Mode.....	13-11
Viewing Reports	13-14
Configuring the Energy Information Platform Reporting Framework for use with Oracle BI Publisher ..	13-15
Oracle BI Publisher Configuration	13-15
Energy Information Platform Configuration	13-18

Adding Oracle BI Publisher Reports to the Energy Information Platform.....	13-19
Designing Oracle BI Publisher Reports for use with the Reporting Framework	13-21
Report Folders.....	13-21
Report Data Sources.....	13-21
Creating Parameters.....	13-21
Adding Custom Crystal Reports to the Oracle Utilities Energy Information Platform.....	13-24
Adding Custom Reports Using Static ASP Pages.....	13-25
Designing Crystal Reports for use with the Reporting Framework	13-27
Report Data Sources.....	13-27
Creating Parameters and Parameter Fields	13-27
Adding Oracle Utilities Rules Language Rate Schedules to the Energy Information Platform.....	13-29
Adding Rate Schedules.....	13-29
Designing Rate Schedules for use with the Reporting Framework	13-30
Reporting on Interval Data.....	13-35
Creating Crystal Reports Templates.....	13-35
Converting Interval Data	13-35
Interval Data Reporting Tables	13-37
Chapter 14	
Troubleshooting.....	14-1
Rules Language Profiling.....	14-2
Running Trial Calculations	14-2
Within an Application	14-2
Sample Profile Output	14-4
SQL Tracing.....	14-5
Enabling SQL Tracing	14-5
Sample SQL Tracing File.....	14-5
Oracle Utilities Energy Information Platform Diagnostics	14-6
Oracle Utilities Energy Information Platform Adapter Logging.....	14-8
Setting the Debug Level.....	14-8
Viewing Log Files	14-8
Debug Level Details and Sample Log Files	14-9
Part Three	
COM Interfaces.....	1-i
Chapter 15	
Energy Information Platform COM Interfaces	15-1
The Energy Information Platform COM Interfaces	15-2
COM Interface Level Functions.....	15-2
Energy Information Platform Level Functions.....	15-3
How the Energy Information Platform COM Interfaces Work.....	15-4
The Energy Information Platform COM Interface Format.....	15-5
Basic Syntax	15-6
Common Argument Definitions.....	15-7
xmlDataSource	15-7
Date Formats	15-8
Date.....	15-8
Date-Time	15-8
Chapter 16	
Data Source Interface	16-1
Methods, Interface, and Syntax	16-2
Argument Definitions	16-4
Input Values	16-5
xmlDataSource	16-5
xmlQueryIn.....	16-7

Return Values	16-9
xmlQueryOut	16-9
Chapter 17	
Energy Information Platform Database Import Interface	17-1
Methods, Interface, and Syntax	17-2
Interface Arguments	17-3
Input Values	17-4
xmlImport	17-4
Return Values	17-6
xmlImportResult	17-6
Chapter 18	
Oracle Utilities Rules Language and Analysis Interface	18-1
Methods, Interface, and Syntax	18-2
Rules Language Interface Methods	18-2
Analysis Interface Methods	18-4
Interface Arguments	18-6
Input Values	18-8
xmlRulesSchedule	18-8
xmlAnalysis	18-11
Return Values	18-15
xmlSaveToXML	18-15
xmlAnalysisResult	18-16
Chapter 19	
Energy Information Platform Work Queues Interface	19-1
Methods, Interfaces, and Syntax	19-2
Interface Arguments	19-5
Input Values	19-6
xmlWQItem XML Elements	19-6
XML Examples - xmlWQItem	19-6
Element Descriptions - xmlWQItem	19-10
Return Values	19-13
xmlWQItemOut	19-13
Chapter 20	
Energy Information Platform Interval Data Interfaces	20-1
LSINTD Interface Methods, and Parameter Descriptions	20-2
IIntDDDataSource Interface	20-2
IIntDRecorderList Interface	20-5
IIntDRecorder Interface	20-6
IIntDChannelList Interface	20-6
IIntDChannel Interface	20-7
IIntDCutSeries Interface	20-8
IIntDCutList Interface	20-8
IIntDCutMessageList Interface	20-8
IIntDCutMessage Interface	20-9
IIntervalList Interface	20-10
IIntDPureCut Interface	20-10
IIntDCut Interface	20-14
IIntDCutValue Interface	20-20
IIntDProvider Interface	20-21
Interval Data Source Providers	20-22
RDBProvider	20-22
BTEProvider	20-22
DATASOURCE Definition	20-23

Sample Application using LSINTD	20-24
INTDVB Compatibility	20-30
Part Four	
Appendices	1-i
Appendix A	
Oracle Utilities Data Repository Database Schema.....	A-1
Energy Information Platform Schema p. 1	A-2
Energy Information Platform Schema p. 2	A-3
Energy Information Platform Schema - Security	A-4
Energy Information Platform Schema - Reporting	A-5
Energy Information Platform Schema - Messaging.....	A-6
Energy Information Platform Schema - Work Queues	A-7
Energy Information Platform Schema - Interval Data Manager	A-8
Energy Information Platform Schema - Adapter.....	A-9
Energy Information Platform Schema - Service Points and Market Participants.....	A-10
Energy Information Platform Schema - Web Services	A-11
Appendix B	
Oracle Utilities Data Repository Schema Import File Format	B-1
Record Formats	B-2
A Note Regarding Dependent Records.....	B-2
Oracle Utilities Import Format (*.imp).....	B-3
Rules.....	B-5
Date Formats	B-5
Oracle Utilities XML Import Format (*.xml)	B-6
DTD - Oracle Utilities XML Import Format.....	B-6
XML Examples - Oracle Utilities XML Import Format	B-6
Element Descriptions - Oracle Utilities XML Import Format.....	B-7
Appendix C	
Oracle Utilities Enhanced Input/Output Interval Data Format	C-1
Enhanced Format.....	C-2
General Field Descriptions.....	C-2
Field Relationships and Requirements.....	C-3
Enhanced Format Details	C-12
Sample Files.....	C-15
Enhanced Direct Input File	C-15
Standard Direct Input File.....	C-15
Units of Measure	C-16
Appendix D	
Oracle Utilities Standard XML Interval Data Format.....	D-1
Standard XML Format	D-2
General Node Descriptions	D-2
General Rules.....	D-3
Node Relationships and Requirements	D-4
Interval Values.....	D-12
XML Standard File Format Example.....	D-14
XML Standard File Format DTD.....	D-15
XML Compact Format.....	D-16
XML Compact File Format Examples	D-16
XML Compact File Format Element and Attribute Descriptions.....	D-17
Interval Values.....	D-19

Appendix E

Oracle Utilities Comma Separated Values (CSV) Interval Data Format E-1

 CSV Format E-2

 General Description E-3

 Document Description Record E-4

 Field Relationships and Requirements..... E-4

 Format Details E-13

 Sample CSV File E-16

 Differences with the .LSE format..... E-16

 Field Delimiters E-16

 Field Types..... E-16

 Document Description Record E-16

 Date, Time, and Decimal Value Representation in Records..... E-16

 Interval Data Records E-16

 Compatibility Issues E-17

Index

What's New

New Features in the Oracle Utilities Energy Information Platform Configuration Guide

This chapter outlines the new features of the 1.6.0.0 release of the Oracle Utilities Energy Information Platform that are documented in this guide.

New Features for Release 1.6.0.0

Feature	Description	For more information, refer to...
Support for Oracle Business Intelligence Publisher	This release includes support for publishing reports using Oracle Business Intelligence Publisher 10.1.3.4.	LSREPORTMONITOR.CFG.XML on page 2-35 Chapter 13: Energy Information Platform Reporting Framework , including: <ul style="list-style-type: none">• Configuring the Energy Information Platform Reporting Framework for use with Oracle BI Publisher on page 13-15• Adding Oracle BI Publisher Reports to the Energy Information Platform on page 13-19• Designing Oracle BI Publisher Reports for use with the Reporting Framework on page 13-21
New Interval Data Converters	This release includes new interval data converters that can be used when importing interval data using the Adapter.	Chapter 6: Setting Up, Configuring, and Running the Energy Information Platform Adapter , including: <ul style="list-style-type: none">• IntDExpConverter on page 6-32
Enhanced Tracing and Logging	This release includes enhancements to the tracing and logging functionality of the Energy Information Platform Logging Framework.	LSLOGGER.CFG.XML on page 2-27 Energy Information Platform Logging Framework on page 11-25

New Features for Release 1.6.1.0

Feature	Description	For more information, refer to...
Utility to set administrator password	The LSSecureInit.exe program has been enhanced to support setting the password for the administrator login ("admin").	Set Administration Login Password on page 12-16
Support for securing Adapter Server and Adapter Monitor communications	Communications between the Adapter Server and Adapter Monitor can now be secured and encrypted via SSL.	Securing and Encrypting the Adapter Server and Monitor on page 6-79

Chapter 14

Troubleshooting

This chapter describes tools available to aid in troubleshooting performance problems and other issues that may arise when installing and configuring Oracle Utilities Energy Information Platform, including:

- **Rules Language Profiling**
- **SQL Tracing**
- **Oracle Utilities Energy Information Platform Diagnostics**
- **Oracle Utilities Energy Information Platform Adapter Logging**

Rules Language Profiling

Rules Language profiling allows administrators to create a “Code Profile” that lists how long (in milliseconds) it takes the Rules Language to execute each line of code in a profiled rate schedule. This can aid in troubleshooting Rules Language performance issues and problems that may arise during configuration and testing of Oracle Utilities software. Rules Language profiling can be performed in one of two ways:

- **Running Trial Calculations**
- **Within an Application**

Important Note

Rules Language profiling has a negative impact on performance, and should ONLY be used when troubleshooting and/or debugging; Rules Language profiling should be DISABLED when running Oracle Utilities software in a production environment.

Running Trial Calculations

The Trial Calculation module (see **Trial Calculations** on page 10-2 in the *Data Manager User's Guide*) includes the ability to create a code profile when running an analysis in Single-Step mode.

How to create a code profile when using Trial Calculation:

1. Open a Single-Step analysis in the Trial Calculation module in either Data Manager or C/S Oracle Utilities Billing Component.
 - a. Select **Analysis->Trial Calculation** in Data Manager.
 - b. Select the Customer/Account, Bill Period, and Rate Schedule as appropriate.
 - b. Set Options and/or Advanced Options as appropriate.
 - b. Click Single-Step to run the analysis in Single-Step mode.

See **How to Run a Trial Calculation:** on page 10-2 and **How to run the Single Step Analysis** on page 10-4 in the *Data Manager User's Guide* and for more details about running a trial calculation in Single-Step mode.

2. When the windows for the Single-Step mode open, select **Single-Step->Run** (or press **F5**).
3. After the analysis is complete, select **Single-Step->Display Profile** (or press **Shift-F6**)/

The code profile will open in a new report window titled “**Performance Statistics**”. See **Sample Profile Output** on page 14-4 for a sample of the contents of this window.

Within an Application

You can also obtain a Rules Language code profile when executing Rules Language within any Oracle Utilities application, including RUNRS.EXE and the Oracle Utilities Rules Language interface, as well as within rate schedules executed by Oracle Utilities products such as Oracle Utilities Energy Information Platform, Oracle Utilities Billing Component, Oracle Utilities Quotations Management, or Oracle Utilities Load Profiling and Settlement.

Rules Language profiling from within an application can be accomplished through use of:

- **SAVE_PROFILE Function**
- **LS_SAVE_PROFILE_FILENAME Identifier**

SAVE_PROFILE Function

The SAVE_PROFILE function saves a Rules Language code profile to a specified directory/file for the rate schedule in which it's used.

See **SAVE_PROFILE Function** on page 14-3 in the *Oracle Utilities Rules Language Reference Guide* for more information about using this function.

LS_SAVE_PROFILE_FILENAME Identifier

The LS_SAVE_PROFILE_FILENAME identifier specifies the path and file name of a text file that contains the code profile of the rate schedule in which the identifier appears.

Note: This identifier takes precedence over the SAVE_PROFILE function. If this identifier is present in a rate schedule (or INCLUDED Rider or Contract) all calls to the SAVE_PROFILE function are ignored.

This identifier can be set in the following ways:

- In the rate schedule itself, in which case the identifier is used in an ASSIGNMENT statement in the rate schedule.
- Passed as an input parameter to the rate schedule. This can be accomplished through use of the “-x” parameter of RUNRS.EXE or by any other means of passing input parameters to a rate schedule.

See **Reserved Identifiers** on page 4-13 in the *Oracle Utilities Rules Language User's Guide* for more information about this identifier.

Rules Language Profiling When Processing Multiple Accounts

It's also possible (and sometimes desirable) to obtain a code profile when processing multiple customers and/or accounts. This allows you to obtain a code profile that lists overall execution time for a rate schedule for a set of customers and/or accounts, rather than just the execution time for processing a single customer or account.

Multiple Accounts Running the Same Rate Schedule

When the same rate schedule is executed for multiple accounts within a single run of an application, the profile for all the accounts will be accumulated and saved to the specified file.

For example, if processing a group of 100 accounts, the time listed for each line of Rules Language code will represent the total execution time for that line for all accounts processed.

Multiple Accounts Running Different Rate Schedules

When different rate schedules are executed for each customer and/or account being processed. Whenever a rate schedule changes, the profile information will be cleared. For example if two rate schedules are run alternatively (the first for Account “A”, the second for Account “B”, the first again for Account “C”, etc.), at the end of processing, the profile information for the last run of the two rates alone will be available.

Cases where a single rate schedule INCLUDEs or CALLs different riders will be treated as a rate schedule change. However, in these cases, the profiles will be saved with a number appended to the end of the file name. For example, suppose a rate schedule creates a code profile file called “PROFILE.TXT.” If the rate schedule is executed twice (for two accounts) and INCLUDEs or CALLs different riders during each execution, the profiles would be saved as “PROFILE.TXT” and “PROFILE_1.TXT.”

Sample Profile Output

The following is an example of a code profile for a portion of a simple billing calculation rate schedule.

Line#	milliseconds	hits	Code
1			// Saved by lprosper on 12/17/2004 15:47:26
2			//Created by: 11/06/2001 L. Prosperi
3			//Rate: RS_1 Residential Service Rate 1
4			//
5			//Load Account Data
6			//
7	0.08	1	ACCOUNT_ID = ACCOUNT.ACCOUNTID;
8	0.01	1	START = BILL_START;
9	0.01	1	STOP = BILL_STOP;
10			//
11			//Cancel Rebill Rider
12	0.00	1	INCLUDE "CANCEL_REBILL_RIDER";
13			// Saved by lprosper on 12/17/2004 15:47:10
14			/** CANCEL_REBILL_RIDER for BX Training*/
15			/* */
16	0.03	1	IF (BILL_TYPE = "CANCEL") OR (BILL_TYPE
17			= "CANCEL/REBILL")
18			THEN
19			CAN.ACCT_NO = ACCOUNT_ID;
20			CAN.KWH = KWH;
21			CAN.TOT_DUE = \$EFFECTIVE_REVENUE;
22			CAN.BILL_TYPE = BILL_TYPE;
23			CAN.BILL_START = BILL_START;
24			CAN.BILL_STOP = BILL_STOP;
25			SAVE CAN TO CIS SECTION "CANCEL";
26			END IF;
27			/* End INCLUDE "CANCEL_REBILL_RIDER";
			(Nesting Level 1) */
			//

SQL Tracing

SQL tracing allows administrators to trace (and capture in a log file) all database queries executed by Oracle Utilities applications. This can aid in troubleshooting performance issues and database-related problems that may arise during configuration and testing of Oracle Utilities software.

Important Note

SQL tracing has a negative impact on performance, and should ONLY be used when troubleshooting and/or debugging; SQL tracing should be DISABLED when running Oracle Utilities software in a production environment.

Enabling SQL Tracing

SQL tracing for all Oracle Utilities applications (including c/s applications, command line programs, and web applications) can be logged using the LSDB.CFG.XML configuration file. This file defines a log file used to capture all ODBC statements (queries) executed by all Oracle Utilities applications. The LSDB.CFG.XML file must be installed in the **C:\LODESTAR\CFG** directory.

Example:

```
<LS_SQL_CONFIG>
  <SQL_TRACE ENABLE="YES" LOG_PARAMS="YES" FILENAME="U:\SQLLOG.LOG"
  APPEND="NO"/>
</LS_SQL_CONFIG>
```

See **LSDB.CFG.XML** on page 2-26 for more information about setting up this file.

Sample SQL Tracing File

The text below is representative of the output of SQL tracing.

```
...
select distinct  TR0804.ACCOUNT.UIDACCOUNT,
TR0804.ACCOUNT.UIDCUSTOMER,  TR0804.ACCOUNT.ACCOUNTID,
TR0804.ACCOUNT.STARTTIME,  TR0804.ACCOUNT.STOPTIME,
TR0804.ACCOUNT.OPCOCODE,  TR0804.ACCOUNT.JURISCODE,
TR0804.ACCOUNT.NAME,  TR0804.ACCOUNT.SIC,
TR0804.ACCOUNT.ACCOUNTSTATUSCODE,  TR0804.ACCOUNT.REVENUECODE,
TR0804.ACCOUNT.BILLINGMODEFLAG,  TR0804.ACCOUNT.PRINTDETAIL,
TR0804.ACCOUNT.FULLDAYBILL,  TR0804.ACCOUNT.PREWINDOW,
TR0804.ACCOUNT.POSTWINDOW,  TR0804.ACCOUNT.EDI,
TR0804.ACCOUNT.REGIONCODE,  TR0804.ACCOUNT.CURRENCYCODE,
TR0804.ACCOUNT.OWNER,  TR0804.ACCOUNT.ACCOUNTTYPECODE,
TR0804.ACCOUNT.UIDPARTY,  TR0804.ACCOUNT.UIDMARKETPARTICIPANT,
TR0804.ACCOUNT.SPCLHANDLECODE,  TR0804.ACCOUNT.LSUSER,
TR0804.ACCOUNT.LSTIME from  TR0804.ACCOUNT where
TR0804.ACCOUNT.ACCOUNTID <= 'BXT01_RES5' order by
TR0804.ACCOUNT.ACCOUNTID

select  count(distinct  TR0804.ACCOUNT.UIDACCOUNT)  from
TR0804.ACCOUNT where  TR0804.ACCOUNT.ACCOUNTID <= 'BXT01_RES5'
...
```

Oracle Utilities Energy Information Platform Diagnostics

One of the most important steps in installing and configuring the Oracle Utilities Energy Information Platform is set up of the LSSECURE.CFG.XML configuration file. This file specifies the database where security information (including user authentication data such as user IDs and passwords) is stored, and is required in order for the Energy Information Platform to function properly. Errors made while setting up this file can result in the user seeing an error message when they attempt to open the Login screen for the Energy Information Platform.

This section describes some of the more common error messages encountered when attempting to login to the Energy Information Platform.

Incorrect DSN in LSSECURE.CFG.XML File

LSDB5107: Unable to select USER record(s): [Microsoft][ODBC Driver Manager]
Data source name not found and no default driver specified 0x9000226a in
(2) /lodestar/ccs/default.asp

Cause of Error: The DSN specified in the <CONNECTSTRING> element in the LSSECURE.CFG.XML file is incorrect.

Remedy: Verify the correct name of the DSN and update the LSSECURE.CFG.XML file accordingly.

Incorrect User ID or Password in LSSECURE.CFG.XML File

LSDB5107: Unable to select USER record(s): [Oracle][ODBC][Ora]ORA-01017:
invalid username/password; logon denied 0x9000226a in
(2) /lodestar/ccs/default.asp

Cause of Error: The User ID (UID) and/or Password (PWD) specified in the <CONNECTSTRING> element in the LSSECURE.CFG.XML file is incorrect.

Remedy: Verify the correct user ID and/or password and update the LSSECURE.CFG.XML file accordingly.

Incorrect Qualifier in LSSECURE.CFG.XML File

LSDB5107: Unable to select USER record(s): [Oracle][ODBC][Ora]ORA-00942:
table or view does not exist 0x9000226a in (2) /lodestar/ccs/default.asp

Cause of Error: The qualifier specified in the <QUALIFIER> element in the LSSECURE.CFG.XML file is incorrect.

Remedy: Verify the correct qualifier and update the LSSECURE.CFG.XML file accordingly.

TNS Configuration (TNSNames.ora) not correctly configured

LSDB5107: Unable to select USER record(s): [Oracle][ODBC][Ora]ORA-12154:
TNS:could not resolve service name 0x9000226a in (2) /lodestar/ccs/default.asp

Cause of Error: Either the TNS Service Name specified for the DSN (on the ODBC Driver Configuration dialog) is incorrect, or there is an error in the TNSNames.ora file for the selected TNS Service Name.

Remedy: Verify the TNSNames.ora file is correct, and select the correct TNS Service Name

Oracle Utilities Energy Information Platform Adapter Logging

The Oracle Utilities Energy Information Platform Adapter (and Oracle Utilities Transaction Management) can be configured to create log files that capture the activity of Adapter business rules and services for diagnostic and troubleshooting purposes. This section provides an overview of the log files produced by the Adapter, including:

- **Setting the Debug Level**
- **Viewing Log Files**
- **Debug Level Details and Sample Log Files**

Setting the Debug Level

The specific data logged and the amount of information captured in Adapter log files are determined by the Debug Level setting in the System Properties table. See **System Properties** on page 11-12 in the *Oracle Utilities Energy Information Platform User's Guide* for information about creating, viewing, and maintaining System Properties records.

Creating log files allows administrators to diagnose problems with Adapter services and business rules, but has a negative effect on performance. For the best performance, set this to 0 (zero). At level 1, output is sent to a log file, but not to the console. Levels 2 and 3 provide additional output in both the log file and the console. Levels 4 and 5 are considered diagnostic modes, and should not be used unless under specific instruction from a technical support specialist. **Debug Level Details and Sample Log Files** on page 14-9 for more information about the specific data captured in the log file for each Debug Level setting.

Viewing Log Files

Adapter log files are created in the **C:\LODESTAR\LTMH\Runtime\Log** directory on the application or web server on which the Adapter software is running.

A separate log file is created for each Runtime Service that is run on the application or web server. The name of each log file is as follows:

<SERVICE_NAME>.log

where:

- <SERVICE_NAME> is the name of the Runtime Service

Log files are appended to each time a service is executed.

All events and actions captured in each log file are time-stamped in the following format:

YYYY/MM/DD HH:MM:SS:mmm - <EVENT>

Example:

2007-05-24 13:48:16.523 - Import_CSV_Demo started.

Debug Level Details and Sample Log Files

The Debug Level on the System Properties table ranges from 0 (zero) to 5, with each higher setting providing additional detail in the log file, and displaying similar information in the Adapter Server console window (see **The Adapter Server** on page 6-76 for more information about starting and running the Adapter server). This section outlines the specific data captured by each Debug Level, and provides sample log files for each setting.

Debug Level 0

At Debug Level 0 (the lowest setting), the Adapter writes no information to the log file (the log file is empty), but displays some information on the Adapter Server console.

At this level, the Adapter console displays the following:

- Service Starting message
2007-05-24 13:48:16.523 - Import_CSV_Demo started.

Debug Level 1

At Debug Level 1, additional detail is sent to the log file, but not to the Adapter Server console. At this level, the Adapter log file contains the following:

- Service Starting message (including process ID)
2007-05-24 13:48:16.523 - Import_CSV_Demo started with process id 1580.
- Initializing Process message
2007-05-24 13:48:16.783 - Initializing process: Import_CSV_Demo
- Output Mechanism message
2007-05-24 13:48:18.366 - Created FileOutputter.
- Initialization Complete message
2007-05-24 13:48:18.366 - Initialization completed.
- Process Started message
2007-05-24 13:48:28.410 - RT_FilePortal.ProcessFile Started
=====
- Applying Business Rule message (if applicable)
2007-05-24 13:48:29.111 - Applying Business Rule.
- Error XML string (if an error occurred)
2008-10-02 11:25:22.640 - <ERROR CODE="100" DESC="BRE: Meter ID BXT00_RES10 does not exist." DATE="2008-10-02T10:25:22"></ERROR>
- Success/Failure message
2007-05-24 13:48:40.537 - The rule returned successfully.
2008-10-02 11:25:22.640 - The rule returned a failure.
- Business Exception/Rollback message (if an error occurred)
2008-10-02 11:25:22.687 - A business exception was encountered. Rolling back.
- Exception message (if an error occurred - contains details of the Java class where the exception occurred)
2008-10-02 11:25:22.687 - Exception:
com.lodestarcorp.portal.PortalException

- Error message (if an error occurred)
2008-10-02 11:25:22.687 - BRE: Meter ID BXT00_RES10 does not exist.
- Process Finished message
2008-10-02 11:25:23.812 - RT_FilePortal.ProcessFile Finished
=====

Sample Log Files - Debug Level 1

Successful:

```
2007-05-24 13:48:16.523 - Import_CSV_Demo started with process id 1580.
2007-05-24 13:48:16.783 - Initializing process: Import_CSV_Demo
2007-05-24 13:48:18.366 - Created FileOutputter.
2007-05-24 13:48:18.366 - Initialization completed.
2007-05-24 13:48:28.410 - RT_FilePortal.ProcessFile Started
=====
2007-05-24 13:48:29.111 - Applying Business Rule.
2007-05-24 13:48:40.537 - The rule returned successfully.
2007-05-24 13:48:40.557 - RT_FilePortal.ProcessFile Finished
=====
```

Error:

```
2008-10-02 11:25:01.343 - Import_CSV_Demo started with process id 1580.
2008-10-02 11:25:01.359 - Initializing process: Import_CSV_Demo
2008-10-02 11:25:01.828 - Created FileOutputter.
2008-10-02 11:25:01.843 - Initialization completed.
2008-10-02 11:25:21.843 - RT_FilePortal.ProcessFile Started
=====
2008-10-02 11:25:22.015 - Applying Business Rule.
2008-10-02 11:25:22.640 - <ERROR CODE="100" DESC="BRE: Meter ID BXT00_RES10
does not exist." DATE="2008-10-02T10:25:22"></ERROR>

2008-10-02 11:25:22.640 - The rule returned a failure.
2008-10-02 11:25:22.687 - A business exception was encountered. Rolling back.
2008-10-02 11:25:22.687 - Exception: com.lodestarcorp.portal.PortalException
2008-10-02 11:25:22.687 - BRE: Meter ID BXT00_RES10 does not exist.
2008-10-02 11:25:23.812 - RT_FilePortal.ProcessFile Finished
=====
```

Debug Level 2

At Debug Level 2, additional details are sent to both the log file and to the Adapter Server console. At this level, the Adapter log file contains the following:

- Service Starting message (including process ID)
2007-05-24 13:28:46.450 - Import_CSV_Demo started with process id 1580.
- Initializing Process message
2007-05-24 13:28:46.731 - Initializing process: Import_CSV_Demo
- Runtime Service Properties
2007-05-24 13:28:47.332 -
=====

	PROPERTIES	
=====		
==		==
==	INPUT_DC_0 =	==
==	com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser	==
==		==
==	INPUT_DC_1 =	==
==	com.lodestarcorp.portal.data.XSLConverter,C:\LODESTAR\LTMH\CSV_METE	==
==		==
==	R_READS.xsl	==
==	INPUT_DC_2 =	==
==	com.lodestarcorp.portal.data.SchemaValidator,C:\LODESTAR\LTMH\Impor	==
==		==
==	t_CSV_Schema.xsd	==
==	OUTPUT_FILE_DIR = C:\Training\Output	==
==	OUTPUT_FILE_SUFFIX = .XML	==
==	OUTPUT_MECHANISM = F	==
==	POLL_DIRECTORY = C:\Training\CSV	==
==	POLL_EXTENSION = txt	==
==	POLL_INTERVAL = 10	==
==	RDL_MEMORY_SIZE = 3	==
==	RULE_NAME = Import_CSV_Demo	==
==	STORAGE_LEVEL = 2	==
==	WQ_ASSIGNED_TO_USERID = lou_p	==
==	WQ_TYPE = ADAPTER	==
=====		
=====		

- Data Conversion Object creation message(s) (if applicable)
2007-05-24 13:28:47.432 - Created data conversion object INPUT_DC_0:
com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser
- Business Rule Initialization message
2007-05-24 13:28:48.263 - Business Rule initialization completed.
- Output Mechanism message (if applicable)
2007-05-24 13:28:48.293 - Created FileOutputter.
- Initialization Complete message
2007-05-24 13:28:48.353 - Initialization completed.
- Process Started message
2007-05-24 13:29:08.512 - RT_FilePortal.ProcessFile Started
=====
- Applying Business Rule message (if applicable)
2007-05-24 13:29:09.283 - Applying Business Rule.

- Mapped values, one for each column mapped by the RDL (if applicable - RDL processing only)*
2007-05-24 13:29:09.934 - METERREAD.METERID = BXT00_RES1
- Script Parsing message (if applicable)*
2007-05-24 13:29:10.525 - Parse Script JSCRIPT
- Error XML string (if an error occurred)*
- Success/Failure message*
2007-05-24 13:29:17.926 - The rule returned successfully.
- Business Exception/Rollback message (if an error occurred)*
- Exception message (if an error occurred - contains details of the Java class where the exception occurred)*
- Error message (if an error occurred)*
- Output creation message (if applicable)
2007-05-24 13:29:17.926 - Output created at
C:\Training\Output\out_19763.XML
- Process Finished message
2007-05-24 13:29:17.936 - RT_FilePortal.ProcessFile Finished
=====

*These lines are repeated for each record within the payload being processed.

Sample Log File - Debug Level 2

```

2007-05-24 13:28:46.450 - Import_CSV_Demo started with process id 1580.
2007-05-24 13:28:46.731 - Initializing process: Import_CSV_Demo
2007-05-24 13:28:47.332 -
=====
=
==                               PROPERTIES                               ==
=====
=
== INPUT_DC_0                     =                                         ==
==   com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser               ==
== INPUT_DC_1                     =                                         ==
==   com.lodestarcorp.portal.data.XSLConverter,C:\LODESTAR\LTMH\CSV_METE ==
==   R_READS.xml                                                           ==
== INPUT_DC_2                     =                                         ==
==   com.lodestarcorp.portal.data.SchemaValidator,C:\LODESTAR\LTMH\Impor ==
==   t_CSV_Schema.xsd                                                      ==
== OUTPUT_FILE_DIR                = C:\Training\Output                     ==
== OUTPUT_FILE_SUFFIX             = .XML                                   ==
== OUTPUT_MECHANISM               = F                                   ==
== POLL_DIRECTORY                 = C:\Training\CSV                       ==
== POLL_EXTENSION                 = txt                                  ==
== POLL_INTERVAL                  = 10                                   ==
== RDL_MEMORY_SIZE                = 3                                   ==
== RULE_NAME                      = Import_CSV_Demo                     ==
== STORAGE_LEVEL                  = 2                                   ==
== WQ_ASSIGNED_TO_USERID          = lou_p                               ==
== WQ_TYPE                        = ADAPTER                               ==
=====
=
2007-05-24 13:28:47.432 - Created data conversion object INPUT_DC_0:
com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser
2007-05-24 13:28:47.702 - Created data conversion object INPUT_DC_1:
com.lodestarcorp.portal.data.XSLConverter

```

```
2007-05-24 13:28:47.852 - Created data conversion object INPUT_DC_2:
com.lodestarcorp.portal.data.SchemaValidator
2007-05-24 13:28:48.263 - Business Rule initialization completed.
2007-05-24 13:28:48.293 - Created FileOutputter.
2007-05-24 13:28:48.353 - Initialization completed.
2007-05-24 13:29:08.512 - RT_FilePortal.ProcessFile Started
=====
2007-05-24 13:29:09.283 - Applying Business Rule.
2007-05-24 13:29:09.934 - METERREAD.METERID = BXT00_RES1
2007-05-24 13:29:09.934 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:29:09.934 - METERREAD.SERIALNO = 12345
2007-05-24 13:29:09.944 - METERREAD.UNINUMBER = 54321
2007-05-24 13:29:09.944 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:29:10.525 - Parse Script JSCRIPT
2007-05-24 13:29:10.525 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:29:10.525 - METERREAD.STOPREADING = 1500
2007-05-24 13:29:10.525 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:29:10.545 - METERREAD.METERID = BXT00_RES2
2007-05-24 13:29:10.545 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:29:10.545 - METERREAD.SERIALNO = 12345
2007-05-24 13:29:10.545 - METERREAD.UNINUMBER = 54321
2007-05-24 13:29:10.545 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:29:10.545 - Parse Script JSCRIPT
2007-05-24 13:29:10.545 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:29:10.555 - METERREAD.STOPREADING = 1500
2007-05-24 13:29:10.555 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:29:10.565 - METERREAD.METERID = BXT00_RES3
2007-05-24 13:29:10.565 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:29:10.575 - METERREAD.SERIALNO = 12345
2007-05-24 13:29:10.575 - METERREAD.UNINUMBER = 54321
2007-05-24 13:29:10.575 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:29:10.575 - Parse Script JSCRIPT
2007-05-24 13:29:10.575 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:29:10.575 - METERREAD.STOPREADING = 1500
2007-05-24 13:29:10.575 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:29:10.595 - METERREAD.METERID = BXT00_RES4
2007-05-24 13:29:10.595 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:29:10.595 - METERREAD.SERIALNO = 12345
2007-05-24 13:29:10.595 - METERREAD.UNINUMBER = 54321
2007-05-24 13:29:10.595 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:29:10.595 - Parse Script JSCRIPT
2007-05-24 13:29:10.665 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:29:10.665 - METERREAD.STOPREADING = 1500
2007-05-24 13:29:10.665 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:29:10.675 - METERREAD.METERID = BXT00_RES5
2007-05-24 13:29:10.775 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:29:10.795 - METERREAD.SERIALNO = 12345
2007-05-24 13:29:10.795 - METERREAD.UNINUMBER = 54321
2007-05-24 13:29:10.795 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:29:10.795 - Parse Script JSCRIPT
2007-05-24 13:29:10.795 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:29:10.805 - METERREAD.STOPREADING = 1500
2007-05-24 13:29:10.805 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:29:10.825 - METERREAD.METERID = BXT00_GS1
2007-05-24 13:29:10.825 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:29:10.825 - METERREAD.SERIALNO = 12345
2007-05-24 13:29:10.825 - METERREAD.UNINUMBER = 54321
2007-05-24 13:29:10.825 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:29:10.836 - Parse Script JSCRIPT
2007-05-24 13:29:10.836 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:29:10.836 - METERREAD.STOPREADING = 1500
2007-05-24 13:29:10.836 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:29:10.846 - METERREAD.METERID = BXT00_GS2
2007-05-24 13:29:10.846 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:29:10.846 - METERREAD.SERIALNO = 12345
2007-05-24 13:29:10.846 - METERREAD.UNINUMBER = 54321
2007-05-24 13:29:10.856 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:29:10.856 - Parse Script JSCRIPT
```

```

2007-05-24 13:29:10.856 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:29:10.856 - METERREAD.STOPREADING = 1500
2007-05-24 13:29:10.856 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:29:17.926 - The rule returned successfully.
2007-05-24 13:29:17.926 - Output created at C:\Training\Output\out_19763.XML
2007-05-24 13:29:17.936 - RT_FilePortal.ProcessFile Finished
=====

```

Debug Level 3

At Debug Level 3, the Adapter log file contains the following:

- Service Starting message (including process ID)
2007-05-24 13:24:09.212 - Import_CSV_Demo started with process id 1580.
- Initializing Process message
2007-05-24 13:24:09.392 - Initializing process: Import_CSV_Demo
- Runtime Service Properties
2007-05-24 13:24:09.983 -
=====


```

=====
==                                     PROPERTIES                                     ==
=====
== INPUT_DC_0                        =                                           ==
==   com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser
==
== INPUT_DC_1                        =                                           ==
==   com.lodestarcorp.portal.data.XSLConverter,C:\LODESTAR\LTMH\CSV_METE
==
==   R_READS.xsl
== INPUT_DC_2                        =                                           ==
==   com.lodestarcorp.portal.data.SchemaValidator,C:\LODESTAR\LTMH\Impor
==
==   t_CSV_Schema.xsd
== OUTPUT_FILE_DIR                   = C:\Training\Output
== OUTPUT_FILE_SUFFIX                = .XML
== OUTPUT_MECHANISM                  = F
== POLL_DIRECTORY                    = C:\Training\CSV
== POLL_EXTENSION                    = txt
== POLL_INTERVAL                     = 10
== RDL_MEMORY_SIZE                   = 3
== RULE_NAME                         = Import_CSV_Demo
== STORAGE_LEVEL                     = 2
== WQ_ASSIGNED_TO_USERID              = lou_p
== WQ_TYPE                           = ADAPTER
=====
=====

```
- Data Conversion Object creation message(s) (if applicable)
2007-05-24 13:24:10.083 - Created data conversion object INPUT_DC_0:
com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser
- Business Rule Initialization Complete message
2007-05-24 13:24:10.874 - Business Rule initialization completed.
- Output Mechanism message (if applicable)
2007-05-24 13:24:10.904 - Created FileOutputter.
- Initialization Complete message
2007-05-24 13:24:10.924 - Initialization completed.

- Process Started message
2007-05-24 13:24:21.099 - RT_FilePortal.ProcessFile Started
=====
- Performing Data Conversion message (if applicable)
2007-05-24 13:24:21.620 - Performing data conversion...
- Data Conversion Saved to message (if applicable)
2007-05-24 13:24:21.620 - Data conversion saved to
C:\LODESTAR\LTMH\Runtime\log\adapter_51136.xml
- Beginning Input Conversion message (if applicable)
2007-05-24 13:24:21.620 - Beginning input conversion.
- Applying Business Rule message (if applicable)
2007-05-24 13:24:21.850 - Applying Business Rule.
- Streaming message (based on xmlstream Business Rule property)
2007-05-24 13:24:22.851 - STREAMING is OFF
- Mapped values, one for each column mapped by the RDL (if applicable - RDL processing only)*
2007-05-24 13:24:22.861 - METERREAD.METERID = BXT00_RES1
- Script Parsing message (if applicable)*
2007-05-24 13:24:22.871 - Parse Script JSCRIPT
- Global Function message (if applicable)*
2007-05-24 13:24:22.871 - BILLDET('KWH')
- Begin Validation Group message (if applicable)*
2007-05-24 13:24:22.871 - ***** Begin Validation Group ***** Meter
- Begin Validation message (if applicable)*
2007-05-24 13:24:22.871 - ***** Begin Validation ***** MeterID
- Validate from message (if applicable - displays type of validation)*
2007-05-24 13:24:22.871 - ***** Validate from SQL *****
- Validation script (if applicable - displays validation script)*
2007-05-24 13:24:22.871 - select meterid from meter where meterid =
'BXT00_RES1'
- Validation Passed/Failed message (if applicable)*
2007-05-24 13:24:22.881 - ***** Validation Passed ***** MeterID
- Validation Group Passed/Failed message (if applicable)*
2007-05-24 13:24:22.881 - ***** Validation Group Passed ***** Meter
- Error XML string (if an error occurred)*
- Success/Failure message*
2007-05-24 13:24:31.374 - The rule returned successfully.
- Business Exception/Rollback message (if an error occurred)*
- Exception message (if an error occurred - contains details of the Java class where the exception occurred)*

- Error message (if an error occurred)*
- Output creation message (if applicable)
2007-05-24 13:24:31.404 - Output created at
C:\Training\Output\out_51137.XML
- Output Conversion Complete message (if applicable)
2007-05-24 13:24:31.404 - Output conversion complete.
- Process Finished message
2007-05-24 13:24:31.414 - RT_FilePortal.ProcessFile Finished
=====

*These lines are repeated for each record within the payload being processed.

Sample Log File - Debug Level 3

```

2007-05-24 13:24:09.212 - Import_CSV_Demo started with process id 1580.
2007-05-24 13:24:09.392 - Initializing process: Import_CSV_Demo
2007-05-24 13:24:09.983 -
=====
=
==                               PROPERTIES                               ==
=====
=
== INPUT_DC_0                    =                                           ==
==   com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser               ==
== INPUT_DC_1                    =                                           ==
==   com.lodestarcorp.portal.data.XSLConverter,C:\LODESTAR\LTMH\CSV_METE   ==
==   R_READS.xsl                                                         ==
== INPUT_DC_2                    =                                           ==
==   com.lodestarcorp.portal.data.SchemaValidator,C:\LODESTAR\LTMH\Impor  ==
==   t_CSV_Schema.xsd                                                    ==
== OUTPUT_FILE_DIR               = C:\Training\Output                     ==
== OUTPUT_FILE_SUFFIX            = .XML                                   ==
== OUTPUT_MECHANISM              = F                                   ==
== POLL_DIRECTORY                = C:\Training\CSV                       ==
== POLL_EXTENSION                = txt                                   ==
== POLL_INTERVAL                 = 10                                  ==
== RDL_MEMORY_SIZE               = 3                                   ==
== RULE_NAME                     = Import_CSV_Demo                  ==
== STORAGE_LEVEL                 = 2                                   ==
== WQ_ASSIGNED_TO_USERID         = lou_p                             ==
== WQ_TYPE                       = ADAPTER                           ==
=====
=
2007-05-24 13:24:10.083 - Created data conversion object INPUT_DC_0:
com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser
2007-05-24 13:24:10.363 - Created data conversion object INPUT_DC_1:
com.lodestarcorp.portal.data.XSLConverter
2007-05-24 13:24:10.504 - Created data conversion object INPUT_DC_2:
com.lodestarcorp.portal.data.SchemaValidator
2007-05-24 13:24:10.874 - Business Rule initialization completed.
2007-05-24 13:24:10.904 - Created FileOutputter.
2007-05-24 13:24:10.924 - Initialization completed.
2007-05-24 13:24:21.099 - RT_FilePortal.ProcessFile Started
=====
2007-05-24 13:24:21.620 - Performing data conversion...
2007-05-24 13:24:21.620 - Data conversion saved to
C:\LODESTAR\LTMH\Runtime\log\adapter_51136.xml
2007-05-24 13:24:21.620 - Beginning input conversion.
2007-05-24 13:24:21.850 - Applying Business Rule.
2007-05-24 13:24:22.851 - STREAMING is OFF
2007-05-24 13:24:22.861 - METERREAD.METERID = BXT00_RES1
2007-05-24 13:24:22.861 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:24:22.861 - METERREAD.SERIALNO = 12345

```

```
2007-05-24 13:24:22.861 - METERREAD.UNINUMBER = 54321
2007-05-24 13:24:22.861 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:24:22.871 - Parse Script JSCRIPT
2007-05-24 13:24:22.871 - BILLDET('KWH')
2007-05-24 13:24:22.871 - Parse Script JSCRIPT
2007-05-24 13:24:22.871 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:24:22.871 - METERREAD.STOPREADING = 1500
2007-05-24 13:24:22.871 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:24:22.871 - ***** Begin Validation Group ***** Meter
2007-05-24 13:24:22.871 - ***** Begin Validation ***** MeterID
2007-05-24 13:24:22.871 - ***** Validate from SQL *****
2007-05-24 13:24:22.871 - select meterid from meter where meterid =
'BXT00_RES1'
2007-05-24 13:24:22.881 - ***** Validation Passed ***** MeterID
2007-05-24 13:24:22.881 - ***** Validation Group Passed ***** Meter
2007-05-24 13:24:22.881 - METERREAD.METERID = BXT00_RES2
2007-05-24 13:24:22.881 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:24:22.891 - METERREAD.SERIALNO = 12345
2007-05-24 13:24:22.891 - METERREAD.UNINUMBER = 54321
2007-05-24 13:24:22.891 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:24:22.891 - Parse Script JSCRIPT
2007-05-24 13:24:22.891 - BILLDET('KWH')
2007-05-24 13:24:22.891 - Parse Script JSCRIPT
2007-05-24 13:24:22.911 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:24:22.911 - METERREAD.STOPREADING = 1500
2007-05-24 13:24:22.911 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:24:22.922 - ***** Begin Validation Group ***** Meter
2007-05-24 13:24:22.992 - ***** Begin Validation ***** MeterID
2007-05-24 13:24:22.992 - ***** Validate from SQL *****
2007-05-24 13:24:22.992 - select meterid from meter where meterid =
'BXT00_RES2'
2007-05-24 13:24:23.002 - ***** Validation Passed ***** MeterID
2007-05-24 13:24:23.052 - ***** Validation Group Passed ***** Meter
2007-05-24 13:24:23.062 - METERREAD.METERID = BXT00_RES3
2007-05-24 13:24:23.062 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:24:23.072 - METERREAD.SERIALNO = 12345
2007-05-24 13:24:23.072 - METERREAD.UNINUMBER = 54321
2007-05-24 13:24:23.072 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:24:23.072 - Parse Script JSCRIPT
2007-05-24 13:24:23.072 - BILLDET('KWH')
2007-05-24 13:24:23.072 - Parse Script JSCRIPT
2007-05-24 13:24:23.072 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:24:23.072 - METERREAD.STOPREADING = 1500
2007-05-24 13:24:23.072 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:24:23.082 - ***** Begin Validation Group ***** Meter
2007-05-24 13:24:23.082 - ***** Begin Validation ***** MeterID
2007-05-24 13:24:23.082 - ***** Validate from SQL *****
2007-05-24 13:24:23.082 - select meterid from meter where meterid =
'BXT00_RES3'
2007-05-24 13:24:23.092 - ***** Validation Passed ***** MeterID
2007-05-24 13:24:23.092 - ***** Validation Group Passed ***** Meter
2007-05-24 13:24:23.102 - METERREAD.METERID = BXT00_RES4
2007-05-24 13:24:23.102 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:24:23.102 - METERREAD.SERIALNO = 12345
2007-05-24 13:24:23.102 - METERREAD.UNINUMBER = 54321
2007-05-24 13:24:23.102 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:24:23.102 - Parse Script JSCRIPT
2007-05-24 13:24:23.112 - BILLDET('KWH')
2007-05-24 13:24:23.112 - Parse Script JSCRIPT
2007-05-24 13:24:23.112 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:24:23.112 - METERREAD.STOPREADING = 1500
2007-05-24 13:24:23.112 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:24:23.112 - ***** Begin Validation Group ***** Meter
2007-05-24 13:24:23.112 - ***** Begin Validation ***** MeterID
2007-05-24 13:24:23.112 - ***** Validate from SQL *****
2007-05-24 13:24:23.112 - select meterid from meter where meterid =
'BXT00_RES4'
2007-05-24 13:24:23.122 - ***** Validation Passed ***** MeterID
```

```
2007-05-24 13:24:23.122 - ***** Validation Group Passed ***** Meter
2007-05-24 13:24:23.122 - METERREAD.METERID = BXT00_RES5
2007-05-24 13:24:23.122 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:24:23.122 - METERREAD.SERIALNO = 12345
2007-05-24 13:24:23.122 - METERREAD.UNINUMBER = 54321
2007-05-24 13:24:23.122 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:24:23.122 - Parse Script JSCRIPT
2007-05-24 13:24:23.122 - BILLDET('KWH')
2007-05-24 13:24:23.122 - Parse Script JSCRIPT
2007-05-24 13:24:23.122 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:24:23.132 - METERREAD.STOPREADING = 1500
2007-05-24 13:24:23.132 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:24:23.132 - ***** Begin Validation Group ***** Meter
2007-05-24 13:24:23.132 - ***** Begin Validation ***** MeterID
2007-05-24 13:24:23.132 - ***** Validate from SQL *****
2007-05-24 13:24:23.132 - select meterid from meter where meterid =
'BXT00_RES5'
2007-05-24 13:24:23.132 - ***** Validation Passed ***** MeterID
2007-05-24 13:24:23.132 - ***** Validation Group Passed ***** Meter
2007-05-24 13:24:23.142 - METERREAD.METERID = BXT00_GS1
2007-05-24 13:24:23.142 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:24:23.142 - METERREAD.SERIALNO = 12345
2007-05-24 13:24:23.142 - METERREAD.UNINUMBER = 54321
2007-05-24 13:24:23.142 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:24:23.142 - Parse Script JSCRIPT
2007-05-24 13:24:23.142 - BILLDET('KWH')
2007-05-24 13:24:23.142 - Parse Script JSCRIPT
2007-05-24 13:24:23.182 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:24:23.182 - METERREAD.STOPREADING = 1500
2007-05-24 13:24:23.182 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:24:23.192 - ***** Begin Validation Group ***** Meter
2007-05-24 13:24:23.192 - ***** Begin Validation ***** MeterID
2007-05-24 13:24:23.202 - ***** Validate from SQL *****
2007-05-24 13:24:23.202 - select meterid from meter where meterid = 'BXT00_GS1'
2007-05-24 13:24:23.202 - ***** Validation Passed ***** MeterID
2007-05-24 13:24:23.212 - ***** Validation Group Passed ***** Meter
2007-05-24 13:24:23.212 - METERREAD.METERID = BXT00_GS2
2007-05-24 13:24:23.222 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:24:23.222 - METERREAD.SERIALNO = 12345
2007-05-24 13:24:23.222 - METERREAD.UNINUMBER = 54321
2007-05-24 13:24:23.222 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:24:23.222 - Parse Script JSCRIPT
2007-05-24 13:24:23.222 - BILLDET('KWH')
2007-05-24 13:24:23.222 - Parse Script JSCRIPT
2007-05-24 13:24:23.222 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:24:23.232 - METERREAD.STOPREADING = 1500
2007-05-24 13:24:23.232 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:24:23.232 - ***** Begin Validation Group ***** Meter
2007-05-24 13:24:23.232 - ***** Begin Validation ***** MeterID
2007-05-24 13:24:23.242 - ***** Validate from SQL *****
2007-05-24 13:24:23.242 - select meterid from meter where meterid = 'BXT00_GS2'
2007-05-24 13:24:23.252 - ***** Validation Passed ***** MeterID
2007-05-24 13:24:23.252 - ***** Validation Group Passed ***** Meter
2007-05-24 13:24:31.374 - The rule returned successfully.
2007-05-24 13:24:31.404 - Output created at C:\Training\Output\out_51137.XML
2007-05-24 13:24:31.404 - Output conversion complete.
2007-05-24 13:24:31.414 - RT_FilePortal.ProcessFile Finished
=====
```

Debug Level 4

At Debug Level 4, the Adapter log file contains the following:

- Service Starting message (including process ID)
2007-05-24 13:23:07.824 - Import_CSV_Demo started with process id 1580.
 - Initializing Process message
2007-05-24 13:23:07.994 - Initializing process: Import_CSV_Demo
 - Runtime Service Properties
2007-05-24 13:23:08.585 -
=====
- ```

=====
====
== PROPERTIES ==
=====
====
== INPUT_DC_0 = ==
== com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser ==
== = ==
== INPUT_DC_1 = ==
== com.lodestarcorp.portal.data.XSLConverter,C:\LODESTAR\LTMH\CSV_METE ==
== = ==
== R_READS.xsl ==
== INPUT_DC_2 = ==
== com.lodestarcorp.portal.data.SchemaValidator,C:\LODESTAR\LTMH\Impor ==
== = ==
== t_CSV_Schema.xsd ==
== OUTPUT_FILE_DIR = C:\Training\Output ==
== OUTPUT_FILE_SUFFIX = .XML ==
== OUTPUT_MECHANISM = F ==
== POLL_DIRECTORY = C:\Training\CSV ==
== POLL_EXTENSION = txt ==
== POLL_INTERVAL = 10 ==
== RDL_MEMORY_SIZE = 3 ==
== RULE_NAME = Import_CSV_Demo ==
== STORAGE_LEVEL = 2 ==
== WQ_ASSIGNED_TO_USERID = lou_p ==
== WQ_TYPE = ADAPTER ==
=====
=====

```
- Data Conversion Object creation message(s) (if applicable)  
2007-05-24 13:23:08.695 - Created data conversion object INPUT\_DC\_0:  
com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser
  - Business Rule Initialization Complete message  
2007-05-24 13:23:09.486 - Business Rule initialization completed.
  - Output Mechanism message (if applicable)  
2007-05-24 13:23:09.546 - Created FileOutputter.
  - Initialization Complete message  
2007-05-24 13:23:09.546 - Initialization completed.
  - Process Started message  
2007-05-24 13:23:29.715 - RT\_FilePortal.ProcessFile Started  
=====
  - Performing Data Conversion message (if applicable)  
2007-05-24 13:23:30.045 - Performing data conversion...

- Data Conversion Saved to message (if applicable)  
2007-05-24 13:23:30.045 - Data conversion saved to  
C:\LODESTAR\LTMH\Runtime\log\adapter\_1751.xml
- Beginning Input Conversion message (if applicable)  
2007-05-24 13:23:30.055 - Beginning input conversion.
- Applying Business Rule message (if applicable)  
2007-05-24 13:23:30.376 - Applying Business Rule.
- Add Script Function message (if applicable)\*\*  
2007-05-24 13:23:31.457 - \*\*\*\*\* Add Script Function \*\*\*\*\*
- Function message (if applicable)\*\*  
2007-05-24 13:23:31.468 - function BILLDET(sBILLDET) {  
var retval;  
switch (sBILLDET) {  
case 'KWH':  
retval='1';  
break;  
}  
return retval;  
}
- Add Script Function Complete message (if applicable)\*\*  
2007-05-24 13:23:31.457 - \*\*\*\*\* Add Script Function Completed \*\*\*\*\*
- Streaming message (based on xmlstream Business Rule property)  
2007-05-24 13:23:31.488 - STREAMING is OFF
- Start Document Event message (if applicable)  
2007-05-24 13:23:31.488 - \*\*\*\*\* Start Document Event \*\*\*\*\*
- Field To XML string (if applicable - defines RDL mappings)\*\*\*  
2007-05-24 13:23:31.488 - <Field to="METERID" tag="/METERREADS/METERREAD/  
METERID" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></  
Field>
- Mapped values, one for each column mapped by the RDL (if applicable - RDL processing only)\*  
2007-05-24 13:23:31.608 - METERREAD.METERID = BXT00\_RES1
- Script Parsing message (if applicable)\*  
2007-05-24 13:23:31.608 - Parse Script JSCRIPT
- Global Function message (if applicable)\*  
2007-05-24 13:23:31.608 - BILLDET('KWH')
- Perform Validations message (if applicable)  
2007-05-24 13:23:31.608 - \*\*\*\*\* Perform Validations \*\*\*\*\* METERREAD
- Begin Validation Group message (if applicable)\*  
2007-05-24 13:23:31.608 - \*\*\*\*\* Begin Validation Group \*\*\*\*\* Meter
- Begin Validation message (if applicable)\*  
2007-05-24 13:23:31.618 - \*\*\*\*\* Begin Validation \*\*\*\*\* MeterID

- Validate from message (if applicable - displays type of validation)\*  
2007-05-24 13:23:31.618 - \*\*\*\*\* Validate from SQL \*\*\*\*\*
- Validation script (if applicable - displays validation script)\*  
2007-05-24 13:23:31.618 - select meterid from meter where meterid = 'BXT00\_RES1'
- Validation Passed/Failed message (if applicable)\*  
2007-05-24 13:23:31.708 - \*\*\*\*\* Validation Passed \*\*\*\*\* MeterID
- Validation Group Passed/Failed message (if applicable)\*  
2007-05-24 13:23:31.708 - \*\*\*\*\* Validation Group Passed \*\*\*\*\* Meter
- Perform Table Action message (if applicable)\*  
2007-05-24 13:23:31.708 - \*\*\*\*\* Perform Table Actions \*\*\*\*\*  
METERREAD
- ROW ACTION XML string (if applicable - contains column values and any errors for each record)\*  
2007-05-24 13:23:31.708 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00\_RES1" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
- End Document Event message (if applicable)  
2007-05-24 13:23:32.028 - \*\*\*\*\* End Document Event \*\*\*\*\*
- ADAPTER XML string (if applicable - contains column values for entire payload)  
2007-05-24 13:23:32.038 - <ADAPTER><LOCALMEMORY><CUT><INTS/></CUT><BABATCH><ADDUPDATE><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00\_RES1" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00\_RES2" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00\_RES3" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00\_RES4" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00\_RES5" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00\_RES6" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW></ADAPTER>
- Error XML string (if an error occurred)
- Success/Failure message\*  
2007-05-24 13:23:40.420 - The rule returned successfully.

- Business Exception/Rollback message (if an error occurred)\*
- Exception message (if an error occurred - contains details of the Java class where the exception occurred)\*
- Error message (if an error occurred)\*
- Output creation message (if applicable)  
2007-05-24 13:23:40.521 - Output created at C:\Training\Output\out\_1752.XML
- Output Conversion Complete message (if applicable)  
2007-05-24 13:23:40.521 - Output conversion complete.
- Process Finished message  
2007-05-24 13:23:40.541 - RT\_FilePortal.ProcessFile Finished  
=====

\*These lines are repeated for each record within the payload being processed.

\*\* These lines are repeated for each Global Function defined, regardless of whether or not the function is used in the RDL

\*\*\* Repeated for each column mapping defined in the RDL

### Sample Log File - Debug Level 4

```

2007-05-24 13:23:07.824 - Import_CSV_Demo started with process id 1580.
2007-05-24 13:23:07.994 - Initializing process: Import_CSV_Demo
2007-05-24 13:23:08.585 -
=====
=
== PROPERTIES ==
=====
=
== INPUT_DC_0 = ==
== com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser ==
== INPUT_DC_1 = ==
== com.lodestarcorp.portal.data.XSLConverter,C:\LODESTAR\LTMH\CSV_METE ==
== R_READS.xml ==
== INPUT_DC_2 = ==
== com.lodestarcorp.portal.data.SchemaValidator,C:\LODESTAR\LTMH\Impor ==
== t_CSV_Schema.xsd ==
== OUTPUT_FILE_DIR = C:\Training\Output ==
== OUTPUT_FILE_SUFFIX = .XML ==
== OUTPUT_MECHANISM = F ==
== POLL_DIRECTORY = C:\Training\CSV ==
== POLL_EXTENSION = txt ==
== POLL_INTERVAL = 10 ==
== RDL_MEMORY_SIZE = 3 ==
== RULE_NAME = Import_CSV_Demo ==
== STORAGE_LEVEL = 2 ==
== WQ_ASSIGNED_TO_USERID = lou_p ==
== WQ_TYPE = ADAPTER ==
=====
=
2007-05-24 13:23:08.695 - Created data conversion object INPUT_DC_0:
com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser
2007-05-24 13:23:08.975 - Created data conversion object INPUT_DC_1:
com.lodestarcorp.portal.data.XSLConverter
2007-05-24 13:23:09.115 - Created data conversion object INPUT_DC_2:
com.lodestarcorp.portal.data.SchemaValidator
2007-05-24 13:23:09.486 - Business Rule initialization completed.
2007-05-24 13:23:09.546 - Created FileOutputter.
2007-05-24 13:23:09.546 - Initialization completed.
2007-05-24 13:23:29.715 - RT_FilePortal.ProcessFile Started
=====

```

```
2007-05-24 13:23:30.045 - Performing data conversion...
2007-05-24 13:23:30.045 - Data conversion saved to
C:\LODESTAR\LTMH\Runtime\log\adapter_1751.xml
2007-05-24 13:23:30.055 - Beginning input conversion.
2007-05-24 13:23:30.376 - Applying Business Rule.
2007-05-24 13:23:31.457 - ***** Add Script Function *****
2007-05-24 13:23:31.457 - function MeterId(sMeter){
sMeter = sMeter + "-MTR";
return sMeter;}
2007-05-24 13:23:31.457 - ***** Add Script Function Completed *****
2007-05-24 13:23:31.457 - ***** Add Script Function *****
2007-05-24 13:23:31.457 - function UOM(sUOM) {
var retval;
switch (sUOM) {
case 'KWH':
retval='01';
break;
case 'KW':
retval='02';
break;
}
return retval;
}
2007-05-24 13:23:31.457 - ***** Add Script Function Completed *****
2007-05-24 13:23:31.468 - ***** Add Script Function *****
2007-05-24 13:23:31.468 - function BILLDET(sBILLDET) {
var retval;
switch (sBILLDET) {
case 'KWH':
retval='1';
break;
}
return retval;
}
2007-05-24 13:23:31.468 - ***** Add Script Function Completed *****
2007-05-24 13:23:31.468 - ***** Add Script Function *****
2007-05-24 13:23:31.468 - function CreateGUID() {
var x = new ActiveXObject("Scriptlet.TypeLib");
return x.GUID.substring(1, 37);
}
2007-05-24 13:23:31.468 - ***** Add Script Function Completed *****
2007-05-24 13:23:31.488 - STREAMING is OFF
2007-05-24 13:23:31.488 - ***** Start Document Event *****
2007-05-24 13:23:31.488 - <Field to="METERID" tag="/METERREADS/METERREAD/
METERID" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></Field>
2007-05-24 13:23:31.488 - <Field to="MANUFACTURER" tag="/METERREADS/METERREAD/
MANUFACTURER" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></
Field>
2007-05-24 13:23:31.488 - <Field to="SERIALNO" tag="/METERREADS/METERREAD/
SERIALNO" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></Field>
2007-05-24 13:23:31.528 - <Field to="UNINUMBER" tag="/METERREADS/METERREAD/
UNINUMBER" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></Field>
2007-05-24 13:23:31.528 - <Field to="METERREADTIME" tag="/METERREADS/METERREAD/
METERREADTIME" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></
Field>
2007-05-24 13:23:31.528 - <Field to="BILLDETERMCODE" tag="/METERREADS/
METERREAD/BILLDET" fmt="" inst="BILLDET" stdout="" attr=""
type="PJScript"><![CDATA['%%THIS%']]></Field>
2007-05-24 13:23:31.608 - <Field to="METERREADMONTH" tag="/METERREADS/
METERREAD/READMONTH" fmt="" inst="" stdout="" attr=""
type="Direct"><![CDATA[]]></Field>
2007-05-24 13:23:31.608 - <Field to="STOPREADING" tag="/METERREADS/METERREAD/
STOPREADING" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></
Field>
2007-05-24 13:23:31.608 - METERREAD.METERID = BXT00_RES1
2007-05-24 13:23:31.608 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:23:31.608 - METERREAD.SERIALNO = 12345
2007-05-24 13:23:31.608 - METERREAD.UNINUMBER = 54321
```

```
2007-05-24 13:23:31.608 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:23:31.608 - Parse Script JSCRIPT
2007-05-24 13:23:31.608 - BILLDET('KWH')
2007-05-24 13:23:31.608 - Run Script BILLDET('KWH')
2007-05-24 13:23:31.608 - BILLDET('KWH')
2007-05-24 13:23:31.608 - Parse Script JSCRIPT
2007-05-24 13:23:31.608 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:23:31.608 - METERREAD.STOPREADING = 1500
2007-05-24 13:23:31.608 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:23:31.608 - ***** Perform Validations ***** METERREAD
2007-05-24 13:23:31.608 - ***** Begin Validation Group ***** Meter
2007-05-24 13:23:31.618 - ***** Begin Validation ***** MeterID
2007-05-24 13:23:31.618 - ***** Validate from SQL *****
2007-05-24 13:23:31.618 - select meterid from meter where meterid =
'BXT00_RES1'
2007-05-24 13:23:31.708 - ***** Validation Passed ***** MeterID
2007-05-24 13:23:31.708 - ***** Validation Group Passed ***** Meter
2007-05-24 13:23:31.708 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:23:31.708 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES1" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:23:31.708 - METERREAD.METERID = BXT00_RES2
2007-05-24 13:23:31.718 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:23:31.718 - METERREAD.SERIALNO = 12345
2007-05-24 13:23:31.718 - METERREAD.UNINUMBER = 54321
2007-05-24 13:23:31.718 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:23:31.718 - Parse Script JSCRIPT
2007-05-24 13:23:31.718 - BILLDET('KWH')
2007-05-24 13:23:31.718 - Run Script BILLDET('KWH')
2007-05-24 13:23:31.718 - BILLDET('KWH')
2007-05-24 13:23:31.718 - Parse Script JSCRIPT
2007-05-24 13:23:31.718 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:23:31.718 - METERREAD.STOPREADING = 1500
2007-05-24 13:23:31.718 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:23:31.718 - ***** Perform Validations ***** METERREAD
2007-05-24 13:23:31.718 - ***** Begin Validation Group ***** Meter
2007-05-24 13:23:31.718 - ***** Begin Validation ***** MeterID
2007-05-24 13:23:31.728 - ***** Validate from SQL *****
2007-05-24 13:23:31.728 - select meterid from meter where meterid =
'BXT00_RES2'
2007-05-24 13:23:31.728 - ***** Validation Passed ***** MeterID
2007-05-24 13:23:31.738 - ***** Validation Group Passed ***** Meter
2007-05-24 13:23:31.738 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:23:31.738 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES2" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:23:31.798 - METERREAD.METERID = BXT00_RES3
2007-05-24 13:23:31.798 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:23:31.808 - METERREAD.SERIALNO = 12345
2007-05-24 13:23:31.808 - METERREAD.UNINUMBER = 54321
2007-05-24 13:23:31.808 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:23:31.808 - Parse Script JSCRIPT
2007-05-24 13:23:31.808 - BILLDET('KWH')
2007-05-24 13:23:31.808 - Run Script BILLDET('KWH')
2007-05-24 13:23:31.818 - BILLDET('KWH')
2007-05-24 13:23:31.818 - Parse Script JSCRIPT
2007-05-24 13:23:31.818 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:23:31.818 - METERREAD.STOPREADING = 1500
2007-05-24 13:23:31.818 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:23:31.818 - ***** Perform Validations ***** METERREAD
2007-05-24 13:23:31.828 - ***** Begin Validation Group ***** Meter
2007-05-24 13:23:31.828 - ***** Begin Validation ***** MeterID
2007-05-24 13:23:31.828 - ***** Validate from SQL *****
2007-05-24 13:23:31.828 - select meterid from meter where meterid =
'BXT00_RES3'
2007-05-24 13:23:31.838 - ***** Validation Passed ***** MeterID
```

```
2007-05-24 13:23:31.848 - ***** Validation Group Passed ***** Meter
2007-05-24 13:23:31.848 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:23:31.848 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES3" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:23:31.858 - METERREAD.METERID = BXT00_RES4
2007-05-24 13:23:31.858 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:23:31.858 - METERREAD.SERIALNO = 12345
2007-05-24 13:23:31.858 - METERREAD.UNINUMBER = 54321
2007-05-24 13:23:31.868 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:23:31.868 - Parse Script JSCRIPT
2007-05-24 13:23:31.868 - BILLDET('KWH')
2007-05-24 13:23:31.878 - Run Script BILLDET('KWH')
2007-05-24 13:23:31.878 - BILLDET('KWH')
2007-05-24 13:23:31.878 - Parse Script JSCRIPT
2007-05-24 13:23:31.878 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:23:31.878 - METERREAD.STOPREADING = 1500
2007-05-24 13:23:31.878 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:23:31.878 - ***** Perform Validations ***** METERREAD
2007-05-24 13:23:31.878 - ***** Begin Validation Group ***** Meter
2007-05-24 13:23:31.888 - ***** Begin Validation ***** MeterID
2007-05-24 13:23:31.888 - ***** Validate from SQL *****
2007-05-24 13:23:31.888 - select meterid from meter where meterid =
'BXT00_RES4'
2007-05-24 13:23:31.898 - ***** Validation Passed ***** MeterID
2007-05-24 13:23:31.898 - ***** Validation Group Passed ***** Meter
2007-05-24 13:23:31.898 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:23:31.898 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES4" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:23:31.898 - METERREAD.METERID = BXT00_RES5
2007-05-24 13:23:31.908 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:23:31.908 - METERREAD.SERIALNO = 12345
2007-05-24 13:23:31.908 - METERREAD.UNINUMBER = 54321
2007-05-24 13:23:31.908 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:23:31.908 - Parse Script JSCRIPT
2007-05-24 13:23:31.908 - BILLDET('KWH')
2007-05-24 13:23:31.908 - Run Script BILLDET('KWH')
2007-05-24 13:23:31.908 - BILLDET('KWH')
2007-05-24 13:23:31.908 - Parse Script JSCRIPT
2007-05-24 13:23:31.908 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:23:31.908 - METERREAD.STOPREADING = 1500
2007-05-24 13:23:31.908 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:23:31.908 - ***** Perform Validations ***** METERREAD
2007-05-24 13:23:31.908 - ***** Begin Validation Group ***** Meter
2007-05-24 13:23:31.908 - ***** Begin Validation ***** MeterID
2007-05-24 13:23:31.908 - ***** Validate from SQL *****
2007-05-24 13:23:31.908 - select meterid from meter where meterid =
'BXT00_RES5'
2007-05-24 13:23:31.918 - ***** Validation Passed ***** MeterID
2007-05-24 13:23:31.918 - ***** Validation Group Passed ***** Meter
2007-05-24 13:23:31.918 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:23:31.918 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES5" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:23:31.918 - METERREAD.METERID = BXT00_GS1
2007-05-24 13:23:31.918 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:23:31.918 - METERREAD.SERIALNO = 12345
2007-05-24 13:23:31.918 - METERREAD.UNINUMBER = 54321
2007-05-24 13:23:31.918 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:23:31.918 - Parse Script JSCRIPT
2007-05-24 13:23:31.918 - BILLDET('KWH')
2007-05-24 13:23:31.928 - Run Script BILLDET('KWH')
2007-05-24 13:23:31.928 - BILLDET('KWH')
2007-05-24 13:23:31.928 - Parse Script JSCRIPT
```

```
2007-05-24 13:23:31.928 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:23:31.928 - METERREAD.STOPREADING = 1500
2007-05-24 13:23:31.928 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:23:31.928 - ***** Perform Validations ***** METERREAD
2007-05-24 13:23:31.928 - ***** Begin Validation Group ***** Meter
2007-05-24 13:23:31.928 - ***** Begin Validation ***** MeterID
2007-05-24 13:23:31.928 - ***** Validate from SQL *****
2007-05-24 13:23:31.928 - select meterid from meter where meterid = 'BXT00_GS1'
2007-05-24 13:23:31.938 - ***** Validation Passed ***** MeterID
2007-05-24 13:23:31.938 - ***** Validation Group Passed ***** Meter
2007-05-24 13:23:31.938 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:23:31.948 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_GS1" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:23:31.998 - METERREAD.METERID = BXT00_GS2
2007-05-24 13:23:31.998 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:23:31.998 - METERREAD.SERIALNO = 12345
2007-05-24 13:23:31.998 - METERREAD.UNINUMBER = 54321
2007-05-24 13:23:31.998 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:23:31.998 - Parse Script JSCRIPT
2007-05-24 13:23:31.998 - BILLDET('KWH')
2007-05-24 13:23:31.998 - Run Script BILLDET('KWH')
2007-05-24 13:23:32.008 - BILLDET('KWH')
2007-05-24 13:23:32.008 - Parse Script JSCRIPT
2007-05-24 13:23:32.008 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:23:32.008 - METERREAD.STOPREADING = 1500
2007-05-24 13:23:32.008 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:23:32.008 - ***** Perform Validations ***** METERREAD
2007-05-24 13:23:32.008 - ***** Begin Validation Group ***** Meter
2007-05-24 13:23:32.018 - ***** Begin Validation ***** MeterID
2007-05-24 13:23:32.018 - ***** Validate from SQL *****
2007-05-24 13:23:32.018 - select meterid from meter where meterid = 'BXT00_GS2'
2007-05-24 13:23:32.028 - ***** Validation Passed ***** MeterID
2007-05-24 13:23:32.028 - ***** Validation Group Passed ***** Meter
2007-05-24 13:23:32.028 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:23:32.028 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_GS2" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:23:32.028 - ***** End Document Event *****
2007-05-24 13:23:32.038 - <ADAPTER><LOCALMEMORY><CUT><INTS></>
CUT><BABATCH><ADDUPDATE><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES1" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW
ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_RES2" /><MANUFACTURER
V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/
30/2004 00:00:00" /><BILLDETERMCODE V="1" /><STOPREADING V="1500" /
><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE"
TBL="METERREAD"><METERID V="BXT00_RES3" /><MANUFACTURER V="METERSRUS" /
><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004
00:00:00" /><BILLDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/
2004" /></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_RES4" /
><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /
><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /><STOPREADING
V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE"
TBL="METERREAD"><METERID V="BXT00_RES5" /><MANUFACTURER V="METERSRUS" /
><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004
00:00:00" /><BILLDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/
2004" /></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_GS1" /
><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /
><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /><STOPREADING
V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE"
TBL="METERREAD"><METERID V="BXT00_GS2" /><MANUFACTURER V="METERSRUS" /><SERIALNO
V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /
><BILLDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></
ROW></ADDUPDATE></BABATCH></LOCALMEMORY></ADAPTER>
```

```

2007-05-24 13:23:40.420 - The rule returned successfully.
2007-05-24 13:23:40.521 - Output created at C:\Training\Output\out_1752.XML
2007-05-24 13:23:40.521 - Output conversion complete.
2007-05-24 13:23:40.541 - RT_FilePortal.ProcessFile Finished
=====

```

## Debug Level 5

At Debug Level 5, the Adapter log file contains the following:

- Service Starting message (including process ID)
 

```

2007-05-24 13:21:15.226 - Import_CSV_Demo started with process id 1580.

```
- Initializing Process message
 

```

2007-05-24 13:21:15.416 - Initializing process: Import_CSV_Demo

```
- Runtime Service Properties
 

```

2007-05-24 13:21:16.167 -
=====
====
== PROPERTIES ==
=====
====
== INPUT_DC_0 = ==
== com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser
==
== INPUT_DC_1 = ==
== com.lodestarcorp.portal.data.XSLConverter,C:\LODESTAR\LTMH\CSV_METE
==
== R_READS.xsl
== INPUT_DC_2 = ==
== com.lodestarcorp.portal.data.SchemaValidator,C:\LODESTAR\LTMH\Impor
==
== t_CSV_Schema.xsd
== OUTPUT_FILE_DIR = C:\Training\Output
== OUTPUT_FILE_SUFFIX = .XML
== OUTPUT_MECHANISM = F
== POLL_DIRECTORY = C:\Training\CSV
== POLL_EXTENSION = txt
== POLL_INTERVAL = 10
== RDL_MEMORY_SIZE = 3
== RULE_NAME = Import_CSV_Demo
== STORAGE_LEVEL = 2
== WQ_ASSIGNED_TO_USERID = lou_p
== WQ_TYPE = ADAPTER
=====
====

```
- Data Conversion Object creation message(s) (if applicable)
 

```

2007-05-24 13:21:16.278 - Created data conversion object INPUT_DC_0:
com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser

```
- Business Rule Initialization Complete message
 

```

2007-05-24 13:21:19.242 - Business Rule initialization completed.

```
- Output Mechanism message (if applicable)
 

```

2007-05-24 13:21:19.372 - Created FileOutputter

```
- Initialization Complete message
 

```

2007-05-24 13:21:19.372 - Initialization completed.

```

- Process Started message  

```
2007-05-24 13:21:39.704 - RT_FilePortal.ProcessFile Started
=====
```
- Performing Data Conversion message (if applicable)  

```
2007-05-24 13:21:42.829 - Performing data conversion...
```
- Data Conversion Saved to message (if applicable)  

```
2007-05-24 13:21:42.849 - Data conversion saved to
C:\LODESTAR\LTMH\Runtime\log\adapter_31323.xml
```
- Beginning Input Conversion message (if applicable)  

```
2007-05-24 13:21:42.849 - Beginning input conversion.
```
- Applying Business Rule message (if applicable)  

```
2007-05-24 13:21:43.049 - Applying Business Rule.
```
- Add Script Function message (if applicable)\*\*  

```
2007-05-24 13:21:44.862 - ***** Add Script Function *****
```
- Function message (if applicable)\*\*  

```
2007-05-24 13:21:44.872 - function BILLDET(sBILLDET) {
 var retval;
 switch (sBILLDET) {
case 'KWH':
 retval='1';
 break;
 }
 return retval;
}
```
- Add Script Function Complete message (if applicable)\*\*  

```
2007-05-24 13:21:44.872 - ***** Add Script Function Completed *****
```
- Streaming message (based on xmlstream Business Rule property)  

```
2007-05-24 13:21:47.887 - STREAMING is OFF
```
- Mode message (Batch or Event)  

```
2007-05-24 13:21:47.917 - MODE = BATCH
```
- Debug Level message  

```
2007-05-24 13:21:47.917 - Debug Level = 5
```
- Start Document Event message (if applicable)  

```
2007-05-24 13:21:47.917 - ***** Start Document Event *****
```
- Field To XML string (if applicable - defines RDL mappings)\*\*  

```
2007-05-24 13:21:47.917 - <Field to="METERID" tag="/METERREADS/METERREAD/
METERID" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></
Field>
```
- XML Parsing messages (if applicable - displays how XML payload is parsed)\*  

```
2007-05-24 13:21:48.007 - /METERREADS
2007-05-24 13:21:48.007 - /METERREADS/METERREAD
2007-05-24 13:21:48.007 - /METERREADS/METERREAD/METERID
```
- Mapped values, one for each column mapped by the RDL (if applicable - RDL processing only)\*  

```
2007-05-24 13:21:48.007 - METERREAD.METERID = BXT00_RES1
```

- Script Parsing message (if applicable)\*  
2007-05-24 13:21:48.027 - Parse Script JSCRIPT
- Global Function message (if applicable)\*  
2007-05-24 13:21:48.027 - BILLDET('KWH')
- Perform Validations message (if applicable)  
2007-05-24 13:21:48.027 - \*\*\*\*\* Perform Validations \*\*\*\*\* METERREAD
- Begin Validation Group message (if applicable)\*  
2007-05-24 13:21:48.027 - \*\*\*\*\* Begin Validation Group \*\*\*\*\* Meter
- Begin Validation message (if applicable)\*  
2007-05-24 13:21:48.027 - \*\*\*\*\* Begin Validation \*\*\*\*\* MeterID
- Validate from message (if applicable - displays type of validation)\*  
2007-05-24 13:21:48.027 - \*\*\*\*\* Validate from SQL \*\*\*\*\*
- Validation script (if applicable - displays validation script)\*  
2007-05-24 13:21:48.027 - select meterid from meter where meterid =  
'BXT00\_RES1'
- Validation Passed/Failed message (if applicable)\*  
2007-05-24 13:21:48.157 - \*\*\*\*\* Validation Passed \*\*\*\*\* MeterID
- Validation Group Passed/Failed message (if applicable)\*  
2007-05-24 13:21:48.157 - \*\*\*\*\* Validation Group Passed \*\*\*\*\* Meter
- Perform Table Action message (if applicable)\*  
2007-05-24 13:21:48.167 - \*\*\*\*\* Perform Table Actions \*\*\*\*\*  
METERREAD
- ROW ACTION XML string (if applicable - contains column values and any errors for each record)\*  
2007-05-24 13:21:48.167 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID  
V="BXT00\_RES1" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /  
><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /  
><BILLDETERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/  
2004" /></ROW>
- End Document Event message (if applicable)  
2007-05-24 13:21:48.758 - \*\*\*\*\* End Document Event \*\*\*\*\*

- ADAPTER XML string (if applicable - contains column values for entire payload)

```
2007-05-24 13:21:48.758 - <ADAPTER><LOCALMEMORY><CUT><INTS></>
CUT><BABATCH><ADDUPDATE><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES1"/><MANUFACTURER V="METERSRUS"/><SERIALNO V="12345"/
><UNINUMBER V="54321"/><METERREADTIME V="07/30/2004 00:00:00"/
><BILLDETERMCODE V="1"/><STOPREADING V="1500"/><METERREADMONTH V="07/2004"/
></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_RES2"/
><MANUFACTURER V="METERSRUS"/><SERIALNO V="12345"/><UNINUMBER V="54321"/
><METERREADTIME V="07/30/2004 00:00:00"/><BILLDETERMCODE V="1"/
><STOPREADING V="1500"/><METERREADMONTH V="07/2004"/></ROW><ROW
ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_RES3"/><MANUFACTURER
V="METERSRUS"/><SERIALNO V="12345"/><UNINUMBER V="54321"/><METERREADTIME
V="07/30/2004 00:00:00"/><BILLDETERMCODE V="1"/><STOPREADING V="1500"/
><METERREADMONTH V="07/2004"/></ROW><ROW ACTION="ADDUPDATE"
TBL="METERREAD"><METERID V="BXT00_RES4"/><MANUFACTURER V="METERSRUS"/
><SERIALNO V="12345"/><UNINUMBER V="54321"/><METERREADTIME V="07/30/2004
00:00:00"/><BILLDETERMCODE V="1"/><STOPREADING V="1500"/><METERREADMONTH
V="07/2004"/></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES5"/><MANUFACTURER V="METERSRUS"/><SERIALNO V="12345"/
><UNINUMBER V="54321"/><METERREADTIME V="07/30/2004 00:00:00"/
><BILLDETERMCODE V="1"/><STOPREADING V="1500"/><METERREADMONTH V="07/2004"/
></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_GS1"/
><MANUFACTURER V="METERSRUS"/><SERIALNO V="12345"/><UNINUMBER V="54321"/
><METERREADTIME V="07/30/2004 00:00:00"/><BILLDETERMCODE V="1"/
><STOPREADING V="1500"/><METERREADMONTH V="07/2004"/></ROW><ROW
ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_GS2"/><MANUFACTURER
V="METERSRUS"/><SERIALNO V="12345"/><UNINUMBER V="54321"/><METERREADTIME
V="07/30/2004 00:00:00"/><BILLDETERMCODE V="1"/><STOPREADING V="1500"/
><METERREADMONTH V="07/2004"/></ROW></ADDUPDATE></BABATCH></LOCALMEMORY></
ADAPTER>
```

- Error XML string (if an error occurred)

- Success/Failure message\*

```
2007-05-24 13:21:58.053 - The rule returned successfully.
```

- Business Exception/Rollback message (if an error occurred)\*
- Exception message (if an error occurred - contains details of the Java class where the exception occurred)\*
- Error message (if an error occurred)\*
- Output creation message (if applicable)

```
2007-05-24 13:21:58.083 - Output created at
C:\Training\Output\out_31324.XML
```

- Output Conversion Complete message (if applicable)

```
2007-05-24 13:21:58.083 - Output conversion complete.
```

- Process Finished message

```
2007-05-24 13:21:58.103 - RT_FilePortal.ProcessFile Finished
=====
```

\*These lines are repeated for each record within the payload being processed.

\*\* These lines are repeated for each Global Function defined, regardless of whether or not the function is used in the RDL

\*\*\* Repeated for each column mapping defined in the RDL

## Sample Log File - Debug Level 5

```

2007-05-24 13:21:15.226 - Import_CSV_Demo started with process id 1580.
2007-05-24 13:21:15.416 - Initializing process: Import_CSV_Demo
2007-05-24 13:21:16.167 -
=====
==
== PROPERTIES ==
=====
=
== INPUT_DC_0 = ==
== com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser ==
== INPUT_DC_1 = ==
== com.lodestarcorp.portal.data.XSLConverter,C:\LODESTAR\LTMH\CSV_METE ==
== R_READS.xml ==
== INPUT_DC_2 = ==
== com.lodestarcorp.portal.data.SchemaValidator,C:\LODESTAR\LTMH\Impor ==
== t_CSV_Schema.xsd ==
== OUTPUT_FILE_DIR = C:\Training\Output ==
== OUTPUT_FILE_SUFFIX = .XML ==
== OUTPUT_MECHANISM = F ==
== POLL_DIRECTORY = C:\Training\CSV ==
== POLL_EXTENSION = txt ==
== POLL_INTERVAL = 10 ==
== RDL_MEMORY_SIZE = 3 ==
== RULE_NAME = Import_CSV_Demo ==
== STORAGE_LEVEL = 2 ==
== WQ_ASSIGNED_TO_USERID = lou_p ==
== WQ_TYPE = ADAPTER ==
=====
=
2007-05-24 13:21:16.278 - Created data conversion object INPUT_DC_0:
com.lodestarcorp.core.xml.parsers.ColumnCountCSVParser
2007-05-24 13:21:17.590 - Created data conversion object INPUT_DC_1:
com.lodestarcorp.portal.data.XSLConverter
2007-05-24 13:21:17.770 - Created data conversion object INPUT_DC_2:
com.lodestarcorp.portal.data.SchemaValidator
2007-05-24 13:21:19.242 - Business Rule initialization completed.
2007-05-24 13:21:19.372 - Created FileOutputter.
2007-05-24 13:21:19.372 - Initialization completed.
2007-05-24 13:21:39.704 - RT_FilePortal.ProcessFile Started
=====
2007-05-24 13:21:42.829 - Performing data conversion...
2007-05-24 13:21:42.849 - Data conversion saved to
C:\LODESTAR\LTMH\Runtime\log\adapter_31323.xml
2007-05-24 13:21:42.849 - Beginning input conversion.
2007-05-24 13:21:43.049 - Applying Business Rule.
2007-05-24 13:21:44.862 - ***** Add Script Function *****
2007-05-24 13:21:44.862 - function MeterId(sMeter){
sMeter = sMeter + "-MTR";
return sMeter;}
2007-05-24 13:21:44.872 - ***** Add Script Function Completed *****
2007-05-24 13:21:44.872 - ***** Add Script Function *****
2007-05-24 13:21:44.872 - function UOM(sUOM) {
var retval;
switch (sUOM) {
case 'KWH':
retval='01';
break;
case 'KW':
retval='02';
break;
}
return retval;
}
2007-05-24 13:21:44.872 - ***** Add Script Function Completed *****
2007-05-24 13:21:44.872 - ***** Add Script Function *****

```

```
2007-05-24 13:21:44.872 - function BILLDET(sBILLDET) {
 var retval;
 switch (sBILLDET) {
 case 'KWH':
 retval='1';
 break;
 }
 return retval;
}
2007-05-24 13:21:44.872 - ***** Add Script Function Completed *****
2007-05-24 13:21:44.872 - ***** Add Script Function *****
2007-05-24 13:21:44.872 - function CreateGUID() {
 var x = new ActiveXObject("Scriptlet.TypeLib");
 return x.GUID.substring(1, 37);
}
2007-05-24 13:21:44.872 - ***** Add Script Function Completed *****
2007-05-24 13:21:47.887 - STREAMING is OFF
2007-05-24 13:21:47.917 - MODE = BATCH
2007-05-24 13:21:47.917 - Debug Level = 5
2007-05-24 13:21:47.917 - ***** Start Document Event *****
2007-05-24 13:21:47.917 - <Field to="METERID" tag="/METERREADS/METERREAD/
METERID" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></Field>
2007-05-24 13:21:47.917 - <Field to="MANUFACTURER" tag="/METERREADS/METERREAD/
MANUFACTURER" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></
Field>
2007-05-24 13:21:47.917 - <Field to="SERIALNO" tag="/METERREADS/METERREAD/
SERIALNO" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></Field>
2007-05-24 13:21:47.927 - <Field to="UNINUMBER" tag="/METERREADS/METERREAD/
UNINUMBER" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></Field>
2007-05-24 13:21:47.927 - <Field to="METERREADTIME" tag="/METERREADS/METERREAD/
METERREADTIME" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></
Field>
2007-05-24 13:21:47.927 - <Field to="BILLDETERMCODE" tag="/METERREADS/
METERREAD/BILLDET" fmt="" inst="BILLDET" stdout="" attr=""
type="PJScript"><![CDATA['%%THIS%']]></Field>
2007-05-24 13:21:48.007 - <Field to="METERREADMONTH" tag="/METERREADS/
METERREAD/READMONTH" fmt="" inst="" stdout="" attr=""
type="Direct"><![CDATA[]]></Field>
2007-05-24 13:21:48.007 - <Field to="STOPREADING" tag="/METERREADS/METERREAD/
STOPREADING" fmt="" inst="" stdout="" attr="" type="Direct"><![CDATA[]]></
Field>
2007-05-24 13:21:48.007 - /METERREADS
2007-05-24 13:21:48.007 - /METERREADS/METERREAD
2007-05-24 13:21:48.007 - /METERREADS/METERREAD/METERID
2007-05-24 13:21:48.007 - METERREAD.METERID = BXT00_RES1
2007-05-24 13:21:48.017 - /METERREADS/METERREAD/MANUFACTURER
2007-05-24 13:21:48.017 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:21:48.017 - /METERREADS/METERREAD/SERIALNO
2007-05-24 13:21:48.017 - METERREAD.SERIALNO = 12345
2007-05-24 13:21:48.017 - /METERREADS/METERREAD/UNINUMBER
2007-05-24 13:21:48.017 - METERREAD.UNINUMBER = 54321
2007-05-24 13:21:48.017 - /METERREADS/METERREAD/METERREADTIME
2007-05-24 13:21:48.017 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:21:48.017 - /METERREADS/METERREAD/BILLDET
2007-05-24 13:21:48.027 - Parse Script JSCRIPT
2007-05-24 13:21:48.027 - BILLDET('KWH')
2007-05-24 13:21:48.027 - Run Script BILLDET('KWH')
2007-05-24 13:21:48.027 - BILLDET('KWH')
2007-05-24 13:21:48.027 - Parse Script JSCRIPT
2007-05-24 13:21:48.027 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:21:48.027 - /METERREADS/METERREAD/STOPREADING
2007-05-24 13:21:48.027 - METERREAD.STOPREADING = 1500
2007-05-24 13:21:48.027 - /METERREADS/METERREAD/READMONTH
2007-05-24 13:21:48.027 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:21:48.027 - ***** Perform Validations ***** METERREAD
2007-05-24 13:21:48.027 - ***** Begin Validation Group ***** Meter
2007-05-24 13:21:48.027 - ***** Begin Validation ***** MeterID
2007-05-24 13:21:48.027 - ***** Validate from SQL *****
```

```
2007-05-24 13:21:48.027 - select meterid from meter where meterid =
'BXT00_RES1'
2007-05-24 13:21:48.157 - ***** Validation Passed ***** MeterID
2007-05-24 13:21:48.157 - ***** Validation Group Passed ***** Meter
2007-05-24 13:21:48.167 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:21:48.167 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES1" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:21:48.167 - /METERREADS/METERREAD
2007-05-24 13:21:48.167 - /METERREADS/METERREAD/METERID
2007-05-24 13:21:48.167 - METERREAD.METERID = BXT00_RES2
2007-05-24 13:21:48.167 - /METERREADS/METERREAD/MANUFACTURER
2007-05-24 13:21:48.167 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:21:48.167 - /METERREADS/METERREAD/SERIALNO
2007-05-24 13:21:48.167 - METERREAD.SERIALNO = 12345
2007-05-24 13:21:48.167 - /METERREADS/METERREAD/UNINUMBER
2007-05-24 13:21:48.167 - METERREAD.UNINUMBER = 54321
2007-05-24 13:21:48.167 - /METERREADS/METERREAD/METERREADTIME
2007-05-24 13:21:48.167 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:21:48.167 - /METERREADS/METERREAD/BILLDET
2007-05-24 13:21:48.167 - Parse Script JSCRIPT
2007-05-24 13:21:48.167 - BILLDET('KWH')
2007-05-24 13:21:48.167 - Run Script BILLDET('KWH')
2007-05-24 13:21:48.167 - BILLDET('KWH')
2007-05-24 13:21:48.177 - Parse Script JSCRIPT
2007-05-24 13:21:48.177 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:21:48.177 - /METERREADS/METERREAD/STOPREADING
2007-05-24 13:21:48.177 - METERREAD.STOPREADING = 1500
2007-05-24 13:21:48.177 - /METERREADS/METERREAD/READMONTH
2007-05-24 13:21:48.177 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:21:48.177 - ***** Perform Validations ***** METERREAD
2007-05-24 13:21:48.177 - ***** Begin Validation Group ***** Meter
2007-05-24 13:21:48.177 - ***** Begin Validation ***** MeterID
2007-05-24 13:21:48.177 - ***** Validate from SQL *****
2007-05-24 13:21:48.177 - select meterid from meter where meterid =
'BXT00_RES2'
2007-05-24 13:21:48.187 - ***** Validation Passed ***** MeterID
2007-05-24 13:21:48.187 - ***** Validation Group Passed ***** Meter
2007-05-24 13:21:48.197 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:21:48.207 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES2" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:21:48.237 - /METERREADS/METERREAD
2007-05-24 13:21:48.237 - /METERREADS/METERREAD/METERID
2007-05-24 13:21:48.237 - METERREAD.METERID = BXT00_RES3
2007-05-24 13:21:48.237 - /METERREADS/METERREAD/MANUFACTURER
2007-05-24 13:21:48.237 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:21:48.237 - /METERREADS/METERREAD/SERIALNO
2007-05-24 13:21:48.237 - METERREAD.SERIALNO = 12345
2007-05-24 13:21:48.237 - /METERREADS/METERREAD/UNINUMBER
2007-05-24 13:21:48.247 - METERREAD.UNINUMBER = 54321
2007-05-24 13:21:48.247 - /METERREADS/METERREAD/METERREADTIME
2007-05-24 13:21:48.247 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:21:48.257 - /METERREADS/METERREAD/BILLDET
2007-05-24 13:21:48.257 - Parse Script JSCRIPT
2007-05-24 13:21:48.257 - BILLDET('KWH')
2007-05-24 13:21:48.257 - Run Script BILLDET('KWH')
2007-05-24 13:21:48.257 - BILLDET('KWH')
2007-05-24 13:21:48.257 - Parse Script JSCRIPT
2007-05-24 13:21:48.257 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:21:48.257 - /METERREADS/METERREAD/STOPREADING
2007-05-24 13:21:48.257 - METERREAD.STOPREADING = 1500
2007-05-24 13:21:48.267 - /METERREADS/METERREAD/READMONTH
2007-05-24 13:21:48.267 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:21:48.287 - ***** Perform Validations ***** METERREAD
2007-05-24 13:21:48.287 - ***** Begin Validation Group ***** Meter
```

```
2007-05-24 13:21:48.287 - ***** Begin Validation ***** MeterID
2007-05-24 13:21:48.297 - ***** Validate from SQL *****
2007-05-24 13:21:48.297 - select meterid from meter where meterid =
'BXT00_RES3'
2007-05-24 13:21:48.307 - ***** Validation Passed ***** MeterID
2007-05-24 13:21:48.307 - ***** Validation Group Passed ***** Meter
2007-05-24 13:21:48.307 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:21:48.317 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES3" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:21:48.317 - /METERREADS/METERREAD
2007-05-24 13:21:48.317 - /METERREADS/METERREAD/METERID
2007-05-24 13:21:48.317 - METERREAD.METERID = BXT00_RES4
2007-05-24 13:21:48.327 - /METERREADS/METERREAD/MANUFACTURER
2007-05-24 13:21:48.327 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:21:48.327 - /METERREADS/METERREAD/SERIALNO
2007-05-24 13:21:48.327 - METERREAD.SERIALNO = 12345
2007-05-24 13:21:48.327 - /METERREADS/METERREAD/UNINUMBER
2007-05-24 13:21:48.327 - METERREAD.UNINUMBER = 54321
2007-05-24 13:21:48.327 - /METERREADS/METERREAD/METERREADTIME
2007-05-24 13:21:48.327 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:21:48.338 - /METERREADS/METERREAD/BILLDET
2007-05-24 13:21:48.338 - Parse Script JSCRIPT
2007-05-24 13:21:48.338 - BILLDET('KWH')
2007-05-24 13:21:48.338 - Run Script BILLDET('KWH')
2007-05-24 13:21:48.338 - BILLDET('KWH')
2007-05-24 13:21:48.338 - Parse Script JSCRIPT
2007-05-24 13:21:48.338 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:21:48.338 - /METERREADS/METERREAD/STOPREADING
2007-05-24 13:21:48.338 - METERREAD.STOPREADING = 1500
2007-05-24 13:21:48.348 - /METERREADS/METERREAD/READMONTH
2007-05-24 13:21:48.348 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:21:48.348 - ***** Perform Validations ***** METERREAD
2007-05-24 13:21:48.358 - ***** Begin Validation Group ***** Meter
2007-05-24 13:21:48.358 - ***** Begin Validation ***** MeterID
2007-05-24 13:21:48.358 - ***** Validate from SQL *****
2007-05-24 13:21:48.358 - select meterid from meter where meterid =
'BXT00_RES4'
2007-05-24 13:21:48.368 - ***** Validation Passed ***** MeterID
2007-05-24 13:21:48.368 - ***** Validation Group Passed ***** Meter
2007-05-24 13:21:48.368 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:21:48.378 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES4" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:21:48.378 - /METERREADS/METERREAD
2007-05-24 13:21:48.378 - /METERREADS/METERREAD/METERID
2007-05-24 13:21:48.378 - METERREAD.METERID = BXT00_RES5
2007-05-24 13:21:48.378 - /METERREADS/METERREAD/MANUFACTURER
2007-05-24 13:21:48.378 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:21:48.378 - /METERREADS/METERREAD/SERIALNO
2007-05-24 13:21:48.378 - METERREAD.SERIALNO = 12345
2007-05-24 13:21:48.378 - /METERREADS/METERREAD/UNINUMBER
2007-05-24 13:21:48.378 - METERREAD.UNINUMBER = 54321
2007-05-24 13:21:48.378 - /METERREADS/METERREAD/METERREADTIME
2007-05-24 13:21:48.378 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:21:48.388 - /METERREADS/METERREAD/BILLDET
2007-05-24 13:21:48.388 - Parse Script JSCRIPT
2007-05-24 13:21:48.388 - BILLDET('KWH')
2007-05-24 13:21:48.388 - Run Script BILLDET('KWH')
2007-05-24 13:21:48.388 - BILLDET('KWH')
2007-05-24 13:21:48.388 - Parse Script JSCRIPT
2007-05-24 13:21:48.388 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:21:48.388 - /METERREADS/METERREAD/STOPREADING
2007-05-24 13:21:48.388 - METERREAD.STOPREADING = 1500
2007-05-24 13:21:48.388 - /METERREADS/METERREAD/READMONTH
2007-05-24 13:21:48.388 - METERREAD.METERREADMONTH = 07/2004
```

```
2007-05-24 13:21:48.388 - ***** Perform Validations ***** METERREAD
2007-05-24 13:21:48.388 - ***** Begin Validation Group ***** Meter
2007-05-24 13:21:48.388 - ***** Begin Validation ***** MeterID
2007-05-24 13:21:48.388 - ***** Validate from SQL *****
2007-05-24 13:21:48.388 - select meterid from meter where meterid =
'BXT00_RES5'
2007-05-24 13:21:48.398 - ***** Validation Passed ***** MeterID
2007-05-24 13:21:48.398 - ***** Validation Group Passed ***** Meter
2007-05-24 13:21:48.398 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:21:48.398 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES5" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:21:48.398 - /METERREADS/METERREAD
2007-05-24 13:21:48.398 - /METERREADS/METERREAD/METERID
2007-05-24 13:21:48.398 - METERREAD.METERID = BXT00_GS1
2007-05-24 13:21:48.398 - /METERREADS/METERREAD/MANUFACTURER
2007-05-24 13:21:48.398 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:21:48.398 - /METERREADS/METERREAD/SERIALNO
2007-05-24 13:21:48.398 - METERREAD.SERIALNO = 12345
2007-05-24 13:21:48.408 - /METERREADS/METERREAD/UNINUMBER
2007-05-24 13:21:48.408 - METERREAD.UNINUMBER = 54321
2007-05-24 13:21:48.408 - /METERREADS/METERREAD/METERREADTIME
2007-05-24 13:21:48.408 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:21:48.408 - /METERREADS/METERREAD/BILLDET
2007-05-24 13:21:48.408 - Parse Script JSCRIPT
2007-05-24 13:21:48.408 - BILLDET('KWH')
2007-05-24 13:21:48.408 - Run Script BILLDET('KWH')
2007-05-24 13:21:48.408 - BILLDET('KWH')
2007-05-24 13:21:48.408 - Parse Script JSCRIPT
2007-05-24 13:21:48.408 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:21:48.408 - /METERREADS/METERREAD/STOPREADING
2007-05-24 13:21:48.408 - METERREAD.STOPREADING = 1500
2007-05-24 13:21:48.408 - /METERREADS/METERREAD/READMONTH
2007-05-24 13:21:48.408 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:21:48.408 - ***** Perform Validations ***** METERREAD
2007-05-24 13:21:48.408 - ***** Begin Validation Group ***** Meter
2007-05-24 13:21:48.408 - ***** Begin Validation ***** MeterID
2007-05-24 13:21:48.418 - ***** Validate from SQL *****
2007-05-24 13:21:48.418 - select meterid from meter where meterid = 'BXT00_GS1'
2007-05-24 13:21:48.428 - ***** Validation Passed ***** MeterID
2007-05-24 13:21:48.428 - ***** Validation Group Passed ***** Meter
2007-05-24 13:21:48.428 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:21:48.438 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_GS1" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDETERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:21:48.578 - /METERREADS/METERREAD
2007-05-24 13:21:48.578 - /METERREADS/METERREAD/METERID
2007-05-24 13:21:48.578 - METERREAD.METERID = BXT00_GS2
2007-05-24 13:21:48.578 - /METERREADS/METERREAD/MANUFACTURER
2007-05-24 13:21:48.578 - METERREAD.MANUFACTURER = METERSRUS
2007-05-24 13:21:48.588 - /METERREADS/METERREAD/SERIALNO
2007-05-24 13:21:48.588 - METERREAD.SERIALNO = 12345
2007-05-24 13:21:48.588 - /METERREADS/METERREAD/UNINUMBER
2007-05-24 13:21:48.588 - METERREAD.UNINUMBER = 54321
2007-05-24 13:21:48.588 - /METERREADS/METERREAD/METERREADTIME
2007-05-24 13:21:48.588 - METERREAD.METERREADTIME = 07/30/2004 00:00:00
2007-05-24 13:21:48.588 - /METERREADS/METERREAD/BILLDET
2007-05-24 13:21:48.588 - Parse Script JSCRIPT
2007-05-24 13:21:48.588 - BILLDET('KWH')
2007-05-24 13:21:48.588 - Run Script BILLDET('KWH')
2007-05-24 13:21:48.588 - BILLDET('KWH')
2007-05-24 13:21:48.598 - Parse Script JSCRIPT
2007-05-24 13:21:48.598 - METERREAD.BILLDETERMCODE = 1
2007-05-24 13:21:48.598 - /METERREADS/METERREAD/STOPREADING
2007-05-24 13:21:48.608 - METERREAD.STOPREADING = 1500
2007-05-24 13:21:48.608 - /METERREADS/METERREAD/READMONTH
```

```
2007-05-24 13:21:48.608 - METERREAD.METERREADMONTH = 07/2004
2007-05-24 13:21:48.658 - ***** Perform Validations ***** METERREAD
2007-05-24 13:21:48.668 - ***** Begin Validation Group ***** Meter
2007-05-24 13:21:48.668 - ***** Begin Validation ***** MeterID
2007-05-24 13:21:48.668 - ***** Validate from SQL *****
2007-05-24 13:21:48.668 - select meterid from meter where meterid = 'BXT00_GS2'
2007-05-24 13:21:48.748 - ***** Validation Passed ***** MeterID
2007-05-24 13:21:48.748 - ***** Validation Group Passed ***** Meter
2007-05-24 13:21:48.748 - ***** Perform Table Actions ***** METERREAD
2007-05-24 13:21:48.758 - <ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_GS2" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDTERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW>
2007-05-24 13:21:48.758 - ***** End Document Event *****
2007-05-24 13:21:48.758 - <ADAPTER><LOCALMEMORY><CUT><INTS/></
CUT><BABATCH><ADDUPDATE><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID
V="BXT00_RES1" /><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER
V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /><BILLDTERMCODE V="1" /
><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW
ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_RES2" /><MANUFACTURER
V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/
30/2004 00:00:00" /><BILLDTERMCODE V="1" /><STOPREADING V="1500" /
><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE"
TBL="METERREAD"><METERID V="BXT00_RES3" /><MANUFACTURER V="METERSRUS" /
><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004
00:00:00" /><BILLDTERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/
2004" /></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_RES4" /
><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /
><METERREADTIME V="07/30/2004 00:00:00" /><BILLDTERMCODE V="1" /><STOPREADING
V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE"
TBL="METERREAD"><METERID V="BXT00_RES5" /><MANUFACTURER V="METERSRUS" /
><SERIALNO V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004
00:00:00" /><BILLDTERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/
2004" /></ROW><ROW ACTION="ADDUPDATE" TBL="METERREAD"><METERID V="BXT00_GS1" /
><MANUFACTURER V="METERSRUS" /><SERIALNO V="12345" /><UNINUMBER V="54321" /
><METERREADTIME V="07/30/2004 00:00:00" /><BILLDTERMCODE V="1" /><STOPREADING
V="1500" /><METERREADMONTH V="07/2004" /></ROW><ROW ACTION="ADDUPDATE"
TBL="METERREAD"><METERID V="BXT00_GS2" /><MANUFACTURER V="METERSRUS" /><SERIALNO
V="12345" /><UNINUMBER V="54321" /><METERREADTIME V="07/30/2004 00:00:00" /
><BILLDTERMCODE V="1" /><STOPREADING V="1500" /><METERREADMONTH V="07/2004" /></
ROW></ADDUPDATE></BABATCH></LOCALMEMORY></ADAPTER>

2007-05-24 13:21:58.053 - The rule returned successfully.
2007-05-24 13:21:58.083 - Output created at C:\Training\Output\out_31324.XML
2007-05-24 13:21:58.083 - Output conversion complete.
2007-05-24 13:21:58.103 - RT_FilePortal.ProcessFile Finished
=====
```

# Part Three

---

## COM Interfaces

Part Three describes the COM interfaces available with the Oracle Utilities Energy Information Platform, and contains the following chapters:

- **Chapter 15: Energy Information Platform COM Interfaces**
- **Chapter 16: Data Source Interface**
- **Chapter 17: Energy Information Platform Database Import Interface**
- **Chapter 18: Oracle Utilities Rules Language and Analysis Interface**
- **Chapter 19: Energy Information Platform Work Queues Interface**
- **Chapter 20: Energy Information Platform Interval Data Interfaces**

---

# Chapter 15

---

## Energy Information Platform COM Interfaces

This chapter provides a broad overview of the Energy Information Platform Component Object Model (COM) interfaces, including a description of how the COM interfaces function, and the format in which the specific COM interface methods/functions are described in later chapters of this manual.

# The Energy Information Platform COM Interfaces

The Energy Information Platform COM interfaces are divided into two layers. The first is a COM Interface level and the second is the functional layer.

## COM Interface Level Functions

The COM Interface level:

1. Accepts calls from the external applications (such as an EAI Adapter),
2. Exposes some of the Energy Information Platform functions as public COM objects,
3. Returns possible errors, and requested data to the calling programs resulting from the execution of these functions,
4. Calls the Energy Information Platform function passing the XML argument sent from the external application. The function will perform the connect and disconnect from the database if required.
5. Other work to satisfy the business event.

## Return Values

An XML structure will be returned to the calling application, from the COM Interface, that contains function execution results. These results will contain one or more of the following:

- Return codes,
- Error codes,
- Descriptions of errors,
- Requested data, and/or
- Other information satisfying the caller's intention.

## Energy Information Platform Level Functions

The Energy Information Platform Level functions/methods are part of several different interfaces, including:

- **DataSource:** The Data Source (IDataSource) interface provides methods related to data source operations, including opening/close database connections, and beginning, committing, and rolling back transactions.
- **Database:** The Database interface (IDatabase) provides methods related to retrieving, updating, and modifying data in the Oracle Utilities Data Repository, including Add/Update/Delete Data, and Interval Data Information Request.
- **Analysis:** The Analysis interface (IAnalysis) provides methods related to billing calculations and analysis, including Cancel/Rebill, Current/Final Bill, Trial Bill, and processing of rate schedules.
- **AREngine:** The AREngine interface (IAREngine) provides methods related to financial management functionality including updating the subledger, journal balancing, and account balancing.
- **Billing:** The Billing interface (IBilling) includes provides related to financial management billing functionality including posting and canceling of charges and bills.
- **Remittance:** The Remittance interface (IRemittance) provides methods related to financial management remittance functionality including posting and processing of payments, batch payments, and payment files.
- **Maintenance:** The Maintenance interface (IMaintenance) provides methods related to financial management maintenance functionality including posting and canceling of adjustments, transfers, write-offs, and refunds.
- **Messaging:** The Messaging interface (IMessaging) provides methods related to messaging functionality including posting, retrieving, removing, listing of messages.
- **Activity Implementations:** The Activity Implementation interface (IActivityImplementation) provides methods for working with activity implementations in WorkflowExpert.
- **Process Versions:** The Process Versions interface (IProcessVersions) provides methods for working with processes and process versions in WorkflowExpert.
- **Process Instance:** The Process Instance interface (IProcessInstance) provides methods for working with process and activity instances in WorkflowExpert.

The specific methods/functions of each interface are described in the Chapters 3 through 14.

## How the Energy Information Platform COM Interfaces Work

All of the Energy Information Platform COM interfaces work in the same manner. The Energy Information Platform COM Interfaces accept an XML argument(s) based on the specific business functions being performed. The Energy Information Platform COM API initially logs on to (or connects to) the database using the data in the `xmlDataSource` parameter. If successful, it then calls an Energy Information Platform function, passing to it the XML string it requires (all Energy Information Platform functions accept XML arguments). The function parses the XML string and, if successful, begins processing the string and processing the data. If the parsing is unsuccessful, the interface will populate an appropriate return parameter with error codes, and a descriptive text string stating that there was an XML parsing error. This information is returned from the Energy Information Platform function. Transactions may encompass the entire job, or may only encompass each Energy Information Platform record in the XML string. If a transaction is unsuccessful, then the COM Layer will populate the appropriate result argument. The return structure will contain descriptive information about why the record(s) erred.

# The Energy Information Platform COM Interface Format

The descriptions of the methods of the Energy Information Platform COM interfaces in this document all follow the same format. This format is as follows:

**Overview:** A brief description and overview of the intended purpose and use of the interface/method.

**Method, Interface, and Syntax:** The method (function) name, interface object, DLL name, Program ID (progid), and syntax for each of the methods available in the interface. This section also includes a brief description of each method.

**Interface Arguments:** Definitions/descriptions of the arguments used by the interface.

**Input Values, including:**

- **Input Data Type Definitions (DTD):** A DTD or XML schema that defines the input data of each argument used by the interface.
- **Input XML Examples:** An example XML string for each argument applicable to the interface.
- **Input Element Descriptions:** Descriptions of the data elements of each argument accepted by the interface.

**Return Values (if any), including:**

- **Return DTD:** A DTD or XML schema that defines the return data of each argument issued by the interface.
- **Return XML:** An example return XML string for each argument applicable to the interface.
- **Return Element Definitions:** Descriptions of the data elements of each argument returned by the interface.

Note that not all methods use all of the above. For instance, many of the Oracle Utilities Receivables Component methods do not have return values.

## Basic Syntax

The basic syntax for Energy Information Platform COM interface methods is as follows:

**Method Name:** <BusinessFunction>

**Interface:** <InterfaceName>

**DLL Name:** <DLL\_NAME.DLL>

**Program ID:** <progid>

**Syntax:**

```
HRESULT <BusinessFunction>(
 [in] BSTR xmlDataSource,
 [in] BSTR xml<BusinessFunction>,
 [in] BSTR other xml arguments*
 [out] BSTR xml<BusinessFunction>Result);
```

- The BusinessFunction is the name of the Energy Information Platform function to satisfy the business event. For example: LODESTAR\_IMPORT.
- The InterfaceName is the name of the interface that contains the method. For example: IAnalysis.
- The DLL Name is the name of the Dynamic-Link Library (DLL) that contains the interface object. For example: LSDB.DLL.
- The Program ID is the program id (progid) of the interface. For example: LSACCT.Billing.
- The xmlDataSource argument is an XML string that contains database connection and other related information (see below for more information).
- The xml<BusinessFunction> argument is an XML string containing all of the information necessary for the execution of the function supporting the business event. The XML and DTDs are defined in the appropriate section of this document for each exposed function. The string 'BusinessFunction' will be replaced by the name of the exposed COM method or function (i.e. xmlLodestarImport).
- In the optional "out" argument, 'BusinessFunction>Result' will reflect the name of the called method, with the string 'Result' concatenated to it (i.e. xmlLodestarImportResult).
- The optional 'other xml arguments' are any other arguments to be passed to the interface in an XML string.

These naming conventions are optional but recommended.

# Common Argument Definitions

## xmlDataSource

All of the Energy Information Platform interfaces use the xmlDataSource argument. The xmlDataSource argument is an XML string that contains database connection and other related information. The Data Type Definition (DTD), an xml example, and element descriptions for the xmlDataSource argument are provided below.

### DTD - xmlDataSource

```
<!DOCTYPE DataSource
[
<!ELEMENT DataSource (Name, ConnectString?, UserID?, Password ?,
 Qualifier?)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT ConnectString (#PCDATA)>
<!ELEMENT UserID (#PCDATA)>
<!ELEMENT Password (#PCDATA)>
<!ELEMENT Qualifier (#PCDATA)>
]>
```

### XML Example - xmlDataSource

```
<DATASOURCE>
 <NAME>LSDATA</NAME>
 <CONNECTSTRING>UID=user1; PWD=user1pass;DSN=LSDATA</CONNECTSTRING>
 <USERID>user1</USERID>
 <PASSWORD>user1pass</PASSWORD>
 <QUALIFIER>PWRLINE</QUALIFIER>
</DATASOURCE>
```

### Element Descriptions - xmlDataSource

Each of the data elements used by the xmlDataSource argument is described below.

**NAME:** The name of the datasource.

**CONNECTSTRING:** The ODBC connect string for the Oracle Utilities Data Repository.

**USERID:** The user ID of the user making the connection to the datasource.

**PASSWORD:** The user's password.

**QUALIFIER:** The qualifier or schema used for database object access (tables, views, etc.).

The only required argument for the DataSource constructors is the data source name. The Qualifier argument provided for each unique data source will become the default for that data source, but can be overridden at any time by explicitly providing them.

## Date Formats

Unless specifically noted otherwise, all Energy Information Platform COM interface methods use the following Date and Date-Time formats:

### Date

Date elements should be in the following format:

**YYYY-MM-DD**

where YYYY is the year, MM is the month, and DD is the day.

### Date-Time

Date-time elements should be in the following format:

**YYYY-MM-DDThh:mm:ss**

where YYYY is the year, MM is the month, DD is the day, hh is the hour, mm is the minute, and ss is the second.

# Chapter 16

---

## Data Source Interface

This chapter describes the methods/functions available to external systems through the Data Source interface (IDataSource). These methods allow users to perform data source operations, such as opening/closing datasource connections from external systems. These functions include the following:

- Open Connection
- Close Connection
- Begin Transaction
- Commit Transaction
- Rollback Transaction
- Execute Query

# Methods, Interface, and Syntax

The methods, interface object, and syntax for the DataSource interface are as follows:

## Open Connection

**Description:** Used to open a persistent database connection in a two-tier client-server (thick or dedicated client) environment.

**Method Name:** OpenConnection

**Interface:** IDataSource

**DLL Name:** LSDB.DLL

**Program ID:** LSDB.DataSource

**Syntax:**

```
HRESULT OpenConnection([in] BSTR xmlDataSource);
```

## Close Connection

**Description:** Used to close a persistent database connection in a two-tier client-server (thick or dedicated client) environment.

**Method Name:** CloseConnection

**Interface:** IDataSource

**DLL Name:** LSDB.DLL

**Program ID:** LSDB.DataSource

**Syntax:**

```
HRESULT CloseConnection([in] BSTR xmlDataSource);
```

## BeginTransaction

**Description:** Used in a two-tier client-server environment to begin a transaction when the client code must encapsulate two or more database access methods into a single transaction.

**Method Name:** BeginTransaction

**Interface:** IDataSource

**DLL Name:** LSDB.DLL

**Program ID:** LSDB.DataSource

**Syntax:**

```
HRESULT BeginTransaction([in] BSTR xmlDataSource);
```

## CommitTransaction

**Description:** Used in a two-tier client-server environment to commit a transaction when the client code must encapsulate two or more database access methods into a single transaction.

**Method Name:** CommitTransaction

**Interface:** IDataSource

**DLL Name:** LSDB.DLL

**Program ID:** LSDB.DataSource

**Syntax:**

```
HRESULT CommitTransaction([in] BSTR xmlDataSource);
```

## RollbackTransaction

**Description:** Used in a two-tier client-server environment to rollback a transaction when the client code must encapsulate two or more database access methods into a single transaction.

**Method Name:** RollbackTransaction

**Interface:** IDataSource

**DLL Name:** LSDB.DLL

**Program ID:** LSDB.DataSource

**Syntax:**

```
HRESULT RollbackTransaction([in] BSTR xmlDataSource);
```

## ExecuteQuery

**Description:** Used to perform one or more commands against the data source. A single command may return records from the data source, insert a new record into the data source, or update or delete an existing record in the data source.

**Method Name:** ExecuteQuery

**Interface:** IDataSource

**DLL Name:** LSDB.DLL

**Program ID:** LSDB.DataSource

**Syntax:**

```
HRESULT ExecuteQuery([in] BSTR xmlDataSource,
 [in] BSTR xmlQueryIn,
 [out, retval] BSTR* xmlQueryOut);
```

## Argument Definitions

The DataSource Interface uses the following arguments:

### **xmlDataSource Argument**

The xmlDataSource argument is an xml string that contains database connection and other related information. A DTD, xml example, and data element descriptions for this argument can be found on page 15-7.

### **xmlQueryIn Argument**

The xmlQueryIn argument is an xml string that specifies a SQL query used to perform one or more commands against the datasource. A DTD, an XML example, and data element descriptions for this argument can be found on page 16-7.

### **xmlQueryOut Argument**

The xmlQueryOut argument is an xml string that contains return values from the Execute Query function. A DTD, an XML example, and data element descriptions for this argument can be found on page 16-9.

# Input Values

The Data Type Definition (DTD), an xml example, and data element descriptions used as input values for the DataSource interface (IDataSource) are provided below.

## xmlDataSource

### DTD - xmlDataSource

```
<!DOCTYPE DataSource
[
<!ELEMENT DataSource (Name, ConnectString?, UserID?, Password ?,
 Qualifier?)>
<!ELEMENT Name (#PCDATA)>
<!ELEMENT ConnectString (#PCDATA)>
<!ELEMENT UserID (#PCDATA)>
<!ELEMENT Password (#PCDATA)>
<!ELEMENT Qualifier (#PCDATA)>
]>
```

### XML Example - xmlDataSource

```
<DATASOURCE>
 <NAME>LSDATA</NAME>
 <CONNECTSTRING>Data Source=TS92;User
ID=ls470bx;Password=password;LSProvider=ODP</CONNECTSTRING>
 <USERID>user1 </USERID>
 <QUALIFIER>PWRLINE</QUALIFIER>
</DATASOURCE>
```

### Element Descriptions - xmlDataSource

Each of the data elements used by the xmlDataSource argument is described below.

**NAME:** The name of the datasource.

**CONNECTSTRING:** The connect string for the Oracle Utilities Data Repository, in one of the following formats:

For Oracle databases:

```
<CONNECTSTRING>Data Source=<data_source>;User
ID=<user_id>;Password=<password>;LSProvider=ODP;</CONNECTSTRING>
```

For Microsoft SQL databases:

```
<CONNECTSTRING>Data Source=<address>;Initial
Catalog=<SQL_database>;User
Id=<user_id>;Password=<password>;MultipleActiveResultSets=True;LSP
rovider=MSSQL;</CONNECTSTRING>
```

where:

<data\_source> is the Oracle TNS Name for the data source, from the TNS\_NAMES.ora file (typically located in the \\<machine>\oracle\network\admin directory)

<user\_id> is the user ID for the database connection

<password> is the password for the supplied user ID.

<address> is the IP address or Hostname of the MS SQL Server database server

<SQL\_database> is the name of the MS SQL Server database

**USERID:** The user ID of the user making the connection to the datasource.

**QUALIFIER:** The qualifier or schema used for database object access (tables, views, etc.).

The only required argument for the DataSource constructors is the data source name. The Qualifier argument provided for each unique data source will become the default for that data source, but can be overridden at any time by explicitly providing them.

## xmlQueryIn

### DTD - xmlQueryIn

```
<!DOCTYPE QUERY
[
<!ELEMENT QUERY (COMMAND+)>
<!ELEMENT COMMAND (REQUEST, RECORDSET?, ERROR?)>
<!ATTLIST COMMAND
 ID CDATA #IMPLIED
 NORECORDSCDATA#IMPLIED
 PAGESIZECDATA#IMPLIED
 PAGENOCDATA #IMPLIED >
<!ELEMENT REQUEST (#PCDATA)>
<!ELEMENT RECORDSET (RECORD*)>
<!ATTLIST RECORDSET
 NUMRECORDSCDATA#IMPLIED>
<!ELEMENT RECORD (EMPTY)>
<!ATTLIST RECORD
 COLNAME1CDATA#IMPLIED
 .
 .
 .
>
<!ELEMENT ERROR (#PCDATA)>
]>
```

### XML Example - xmlQueryIn

```
<QUERY>
 <COMMAND ID="" NORECORDS="FALSE" PAGESIZE="128" PAGENO="1">
 <REQUEST>select accountid,uidaccount,customerid,uidcustomer from
PWRLINE.ACCOUNT,pwrline.customer where ACCOUNTID like '%RM%';</REQUEST>
 </COMMAND>
</QUERY>
```

### Element Descriptions - xmlQueryIn

The use of each individual attribute and element in the xmlQueryIn argument is described below.

**QUERY:** The query that contains one or more command elements.

**COMMAND:** The command element(s) that contains specific attributes of the command

Attributes:

**ID:** Optional ID (used to allow for easy identification of the command on return),

**NORECORDS:** Optional. If set to TRUE, the command will not return any records.

**PAGESIZE:** Optional. Used with PAGENO to specify a limited number of records to be returned starting at a specified page number given the specified page size (typically the SQL "order by" clause should be used when paging). Regardless of whether paging is used or not, a single command is limited to returning a maximum of 128 records..

**PAGENO:** Optional. Used with PAGESIZE to specify a limited number of records to be returned starting at a specified page number given the specified page size (typically the SQL "order by" clause should be used when paging). Regardless of whether paging is used or not, a single command is limited to returning a maximum of 128 records.

Elements

**REQUEST:** The actual text of the command.

**RECORDSET:** Optional. Used to return any requested records.

Attributes:

**NUMRECORDS:** Indicates how many records are returned.

**RECORD:** Individual record in the recordset.

Attributes:

**COLNAME:** Name of non-null column in the returned record.

**ERROR:** Description of any error(s) that occurred while processing the command.

---

# Return Values

## xmlQueryOut

### DTD - xmlQueryOut

Same as the xmlQueryIn argument.

### \XML Example - xmlQueryOut

```
<QUERY>
 <COMMAND ID="" NORECORDS="FALSE" PAGESIZE="128" PAGENO="1">
 <REQUEST>select accountid,uidaccount,customerid,uidcustomer from
PWRLINE.ACCOUNT,pwrline.customer where ACCOUNTID like '%RM%';</REQUEST>
 <RECORDSET NUMRECORDS="1">
 <RECORD ACCOUNTID="RM_1001" UIDACCOUNT= "487" CUSTOMERID= "RM_1"
UIDCUSTOMER= "344">
 </RECORD>
 </RECORDSET>
 </COMMAND>
</QUERY>
```

### Element Descriptions - xmlQueryOut

Same as the xmlQueryIn argument.



# Chapter 17

---

## Energy Information Platform Database Import Interface

This chapter describes a method/function available to external systems through the Energy Information Platform Database Import interface (LSImportDB). This method allows users to import relational data into the Oracle Utilities Data Repository.

## Methods, Interface, and Syntax

The methods, interface objects, and syntax for the Energy Information Platform Database Import interface are as follows:

### Import Data

**Description:** Used to add, update, and/or delete data in the Oracle Utilities Data Repository.

**Method Name:** Import

**Interface:** LSIImportDB

**DLL Name:** LSIImportDB.dll

**Program ID:** LSIImportDB.LSIImportDB

**Syntax:**

```
HRESULT Import ([in] BSTR xmlDataSource,
 [in] VARIANT xmlImport,
 [out] VARIANT xmlImportResult);
```

## Interface Arguments

The Energy Information Platform Database Import interface uses the following arguments:

### **xmlDataSource Argument**

The xmlDataSource argument is an xml string that contains database connection and other related information. A DTD, xml example, and data element descriptions for this argument can be found on page 15-7.

### **xmlImport Argument**

The xmlImport argument contains the appropriate data to be added/updated/deleted. This may be a URL (String/BSTR), a Request object (in an ASP page), an IStream, SAFEARRAY of bytes (VT\_ARRAY|VT\_UI1), a DOMDocument object, or any object that supports IStream, ISequentialStream, or IPersistStream. A DTD, xml examples, and data element descriptions for this argument can be found on page 17-4.

### **xmlImportResult Argument**

The xmlImportResult argument is an xml string that contains return values from the Import function. This argument is a superset of the xmlImport argument. The records that were successful will be removed from the XML. All the records that error out will be present in the output XML along with the error message. An xml example, and data element descriptions for this argument can be found on pages 17-6.

# Input Values

The Data Type Definition (DTD), xml examples, and data element descriptions used as input values for the Energy Information Platform Database interface (LSImportDB) are provided below.

## xmlImport

### DTD - xmlImport

```
<!DOCTYPE LODESTAR_IMPORT
[
<!ELEMENT LODESTAR_IMPORT (ROWS*)>
<!ATTLIST LODESTAR_IMPORT
 ACTION#PCDATA,
 BATCHSIZE#PCDATA,
 ONERROR#PCDATA>
<!ELEMENT ROWS(ROW*)>
<!ATTLIST ROWS
 TBL #PCDATA>
<!ELEMENT ROW (DBCOLUMNNAME)>
<!ATTLIST ROW
 TBL CDATA #IMPLIED,
 ACTION(ADD|UPDATE|DELETE|ADDUPDATE)#IMPLIED>
<!ELEMENT DBCOLUMNNAME(#PCDATA)>
<!ATTLIST DBCOLUMNNAME
 V CDATA "column value",
 NV CDATA "new column value">
]>
```

### XML Examples - xmlImport

#### Add Example

```
<LODESTAR_IMPORT ACTION="ADD" BATCHSIZE="50" ONERROR="CONTINUE">
 <ROWS>
 <ROW TBL="ACCOUNT">
 <CUSTOMERID V="Cust1"/>
 <ACCOUTNID V="Acct1"/>
 <STARTTIME V="01/01/2000 00:00:00"/>
 </ROW>
 </ROWS>
</LODESTAR_IMPORT>
```

#### Update Example

```
<LODESTAR_IMPORT ACTION="UPDATE" BATCHSIZE="50" ONERROR="STOP">
 <ROWS TBL="ACCOUNT">
 <ROW>
 <CUSTOMERID V="Cust1"/>
 <ACCOUTNID V="Acct1" NV="Acct100"/>
 <STARTTIME V="01/01/2001 00:00:00"/>
 </ROW>
 </ROWS>
</LODESTAR_IMPORT>
```

## Element Descriptions - xmlImport

**LODESTAR\_IMPORT:** Root element containing one or more sets of records, each defined in a ROWS element.

Attributes:

**ACTION:** Defines the action performed. Possible values are:

- **ADD:** Adds one or more records to the database. It is an error if the record(s) is already present.
- **UPDATE:** Updates one or more existing records in the database. It is an error if the record does not exist.
- **DELETE:** Deletes one or more records in the database. It is an error if the record to be deleted does not exist.
- **ADDUPDATE:** Adds or Updates one or more records in the database. This is the Default behavior. If a record is present, the function updates it. If the record does not exist, the function adds the record.

**BATCHSIZE:** The number of records to be sent to the database in a single call. Larger numbers generally improve performance. The default is 50. The maximum is 500.

**ONERROR:** Defines how errors are handled. Possible values include:

- **CONTINUE:** Continue importing records after error.
- **STOP:** Stop importing records when an error occurs.

Elements:

**ROWS:** Element containing one or more records for import, each defined in a ROW element.

Attributes:

**TBL:** Table name for all the records defined in the ROW elements included in the ROW element.

Elements:

**ROW:** Element containing one record for import.

Attributes:

**TBL:** Table name for the record. If present in a ROW element, this overrides the TBL attribute on the ROWS element.

Elements:

**DBCOLUMNNAME:** Actual database name of the column. For example, when importing records in the Account ID column of the Account table, this would be "ACCOUNTID".

Attributes:

**V:** Column value. If not present, NULL is assumed. If the value is an XML, it will be specified as a child of the DBCOLUMNNAME element.

**NV:** New column value for the record. Must be specified if updating a key column.

**Notes:**

- The "NV" (New value) attribute is only required for identity columns that need to be updated.
- Also, for "DELETE" operations, providing just the identity columns is sufficient.
- Any missing "V" (Value) attribute will be treated as a NULL value for the column.

## Return Values

This section describes the data returned from the Energy Information Platform Database interface.

### xmlImportResult

#### XML Example - xmlImportResult

```
<LODESTAR_IMPORT ACTION="ADD" BATCHSIZE="50" ONERROR="CONTINUE">
 <ROWS_IMPORTED>10</ROWS_IMPORTED>
 <ROWS_FAILED>1</ROWS_FAILED>
 <ROWS>
 <ROW TBL="ACCOUNT" ERROR="Unable to find customer Cust1">
 <CUSTOMERID V="Cust1"/>
 <ACCOUNTNID V="Acct1"/>
 <STARTTIME V="01/01/2000 00:00:00"/>
 </ROW>
 </ROWS>
</LODESTAR_IMPORT>
```

#### Element Descriptions - xmlImportResult

The xmlImportResult argument contains the same elements as the xmlImport argument, with the following additions:

**ROWS\_IMPORTED:** The number of records successfully imported.

**ROWS\_FAILED:** The number of records that failed import.

**ROWS:** Element containing one or more records for import, each defined in a ROW element (same as for xmlImport).

Elements:

**ROW:** Element containing a record that failed import.

Attributes:

**TBL:** Table name for the record that failed import.

**ERROR:** Error message explaining why the record failed import.

# Chapter 18

---

## Oracle Utilities Rules Language and Analysis Interface

This chapter describes the methods/functions available to external systems through the Oracle Utilities Rules Language and Analysis interfaces. These methods allow users to run pre-defined, pre-existing rate schedules and to execute current/final and bill correction processing.

# Methods, Interface, and Syntax

## Rules Language Interface Methods

The methods, interface objects, and syntax for the Rules Language interface are as follows:

### Run Rate

**Description:** Used to synchronously run a pre-defined rate schedule.

**Method Name:** RunRate

**Interface:** RunRs

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.RunRs

**Syntax:**

```
HRESULT RunRate ([in] BSTR xmlRulesSchedule,
 [retval] BSTR xmlSaveToXML);
```

### Run Rate Queued

**Description:** Used to asynchronously run a pre-defined rate schedule. Rate schedules run using this method are queued, and executed by the Energy Information Platform Report Framework.

**Method Name:** RunRateQueued

**Interface:** RunRs

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.RunRs

**Syntax:**

```
HRESULT RunRateQueued ([in] BSTR xmlRulesSchedule,
 [retval] BSTR GUID);
```

### Run Rate Status

**Description:** Used to obtain the status of a rate schedule initiated by the Run Rate Queued method.

**Method Name:** RunRateStatus

**Interface:** RunRs

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.RunRs

**Syntax:**

```
HRESULT RunRateStatus ([in] BSTR xmlDataSource,
 [in] BSTR GUID);
[retval] BSTR Status);
```

### Run Rate Result

**Description:** Used to obtain results from a rate schedule initiated by the Run Rate Queued method. This method can only be used with rate schedules with a Status of “DONE”.

**Method Name:** RunRateResult

**Interface:** RunRs

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.RunRs

**Syntax:**

```
HRESULT RunRateResult ([in] BSTR xmlDataSource,
 [in] BSTR GUID);
[retval] BSTR xmlSaveToXML);
```

## Run Rate Stop

**Description:** Used to terminate an already running rate schedule initiated by the Run Rate Queued method.

**Method Name:** RunRateStop

**Interface:** RunRs

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.RunRs

**Syntax:**

```
HRESULT RunRateStop ([in] BSTR xmlDataSource,
 [in] BSTR GUID);
```

## Run Rate Error

**Description:** Used to obtain the error message from rate schedules initiated by the Run Rate Queued method with a Status of “ERROR”.

**Method Name:** RunRateError

**Interface:** RunRs

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.RunRs

**Syntax:**

```
HRESULT RunRateError ([in] BSTR xmlDataSource,
 [in] BSTR GUID,
 [retval] BSTR Error);
```

## Analysis Interface Methods

The methods, interface objects, and syntax for the Analysis interface are as follows:

### Run Analysis

**Description:** Used to synchronously execute billing in either Current/Final or Cancel/Rebill (bill correction) modes.

**Method Name:** RunAnalysis

**Interface:** Analysis

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.Analysis

**Syntax:**

```
HRESULT RunRate ([in] BSTR xmlAnalysis,
 [retval] BSTR xmlAnalysisResult);
```

### Run Analysis Queued

**Description:** Used to asynchronously execute billing in either Current/Final or Cancel/Rebill (bill correction) modes. Billing processes run using this method are queued, and executed by the Energy Information Platform Report Framework.

**Method Name:** RunAnalysisQueued

**Interface:** Analysis

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.Analysis

**Syntax:**

```
HRESULT RunRateQueued ([in] BSTR xmlAnalysis,
 [retval] BSTR GUID);
```

### Run Analysis Status

**Description:** Used to obtain the status of a billing process initiated by the Run Analysis Queued method.

**Method Name:** RunAnalysisStatus

**Interface:** Analysis

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.Analysis

**Syntax:**

```
HRESULT RunRateStatus ([in] BSTR xmlDataSource,
 [in] BSTR GUID);
[retval] BSTR Status);
```

## Run Analysis Result

**Description:** Used to obtain results from a billing process initiated by the Run Analysis Queued method. This method can only be used with billing processes with a Status of “DONE”.

**Method Name:** RunAnalysisResult

**Interface:** Analysis

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.Analysis

**Syntax:**

```
HRESULT RunRateResult ([in] BSTR xmlDataSource,
 [in] BSTR GUID);
[retval] BSTR xmlAnalysisResult);
```

## Run Analysis Stop

**Description:** Used to terminate an already running billing process initiated by the Run Analysis Queued method.

**Method Name:** RunAnalysisStop

**Interface:** Analysis

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.Analysis

**Syntax:**

```
HRESULT RunRateStop ([in] BSTR xmlDataSource,
 [in] BSTR GUID);
```

## Run Analysis Error

**Description:** Used to obtain the error message from rbilling processes initiated by the Run Analysis Queued method with a Status of “ERROR”.

**Method Name:** RunAnalysisError

**Interface:** Analysis

**DLL Name:** LSRF.DLL

**Program ID:** LSRF.Analysis

**Syntax:**

```
HRESULT RunRateError ([in] BSTR xmlDataSource,
 [in] BSTR GUID,
 [retval] BSTR Error);
```

## Interface Arguments

The functions/methods of the Oracle Utilities Rules Language and Analysis interfaces use the following arguments:

### **xmlDataSource Argument**

The xmlDataSource argument is an xml string that contains database connection and other related information. A DTD, xml example, and data element descriptions for this argument can be found on page 15-7 in the *Oracle Utilities Energy Information Platform Configuration Guide*.

### **xmlRulesSchedule Argument**

The xmlRulesSchedule argument is an xml string that specifies the parameters necessary to run a rate schedule. The elements map directly to “-” parameters for the RUNRS executable file. Details on these settings may be found in **Chapter 8: Setting Up Processing to run in Batch Mode** in the *Oracle Utilities Energy Information Platform Configuration Guide*. The structure is designed to support multiple schedule requests. A DTD, xml example, and data element descriptions for this argument can be found on page 18-8.

### **xmlAnalysis Argument**

The xmlAnalysis argument is an xml string that specifies the parameters necessary to run billing processing. The elements map directly to “-” parameters for the CURFINAL and BILLCOR executable files. Details on these settings may be found in **Chapter 8: Setting Up Processing to run in Batch Mode** in the *Oracle Utilities Energy Information Platform Configuration Guide*. The structure is designed to support multiple schedule requests. An xml schema, xml example, and data element descriptions for this argument can be found on page 18-11.

### **xmlSaveToXML Argument**

The xmlSaveToXML argument is an xml string that contains results from any SAVE TO XML statements in the rate schedule. A DTD, xml example, and data element descriptions for this argument can be found on page 18-15.

### **xmlAnalysisResult Argument**

The xmlAnalysisResult argument is an xml string that contains results from billing processing. An xml schema, xml example, and data element descriptions for this argument can be found on page 18-16.

### **GUID**

A Unique string identifier for the rate schedule or billing process that was queued. This is returned from the RunRateQueued and RunAnalysisQueued methods, and is an input for the RunRateStatus, RunRateResults, RunRateStop, RunRateError, RunAnalysisStatus, RunAnalysisResults, RunAnalysisStop, and RunAnalysisError methods.

---

## Status

The status of a queued rate schedule or billing process. Will be one of the following:

Status Value	Description
SUSPENDED	The process is suspended
QUEUED	The process is waiting to be started.
INIT	Generation application is in an initialization phase.
PROCESS	The process is currently executing.
DONE	The process is fully completed.
DELETING	The process is waiting to be deleted.
ERROR	The process cannot be completed because of an error.

## Error

The error returned from a rate schedule or billing process whose Status is “ERROR”.

# Input Values

The input values for the methods/functions of the Rules Language and Analysis interfaces (RunRs and Analysis) are provided below.

## xmlRulesSchedule

### DTD - xmlRulesSchedule

```
<!DOCTYPE LODESTAR_RUNSCHEDULE
[
 <!ELEMENT LODESTAR_RUNSCHEDULE (DATASOURCE, CONFIG_FILE_NAME?,
 TIMEOUT?, USERID, SCHEDULE_ARGUMENTS)>
 <!ELEMENT DATASOURCE (NAME, CONNECTSTRING, USERID?, PASSWORD?, QUALIFIER?)>
 <!ELEMENT NAME (#PCDATA)>
 <!ELEMENT CONNECTSTRING (#PCDATA)>
 <!ELEMENT USERID (#PCDATA)>
 <!ELEMENT PASSWORD (#PCDATA)>
 <!ELEMENT QUALIFIER (#PCDATA)>
 <!ELEMENT CONFIG_FILE_NAME (#PCDATA)>
 <!ELEMENT TIMEOUT(#PCDATA)>
 <!ELEMENT USERID(#PCDATA)>
 <!ELEMENT SCHEDULE_ARGUMENTS (INTERVAL_DATABASE_NAME?,
 RATE_FORM_IDENTIFIER, START_DATE, STOP_DATE?, SAVE_RESULTS?,
 (ACCOUNT | CUSTOMER)?, INTERVAL_DATA_ERROR?, RSID_VALUES?)>
 <!ELEMENT INTERVAL_DATABASE_NAME (#PCDATA)>
 <!ELEMENT RATE_FORM_IDENTIFIER (#PCDATA)>
 <!ELEMENT START_DATE (#PCDATA)>
 <!ELEMENT STOP_DATE (#PCDATA)>
 <!ELEMENT SAVE_RESULTS EMPTY>
 <!ELEMENT ACCOUNT (ALL | ID | LIST_NAME)>
 <!ELEMENT CUSTOMER (ALL | ID | LIST_NAME)>
 <!ELEMENT ALL EMPTY>
 <!ELEMENT ID (#PCDATA)>
 <!ELEMENT LIST_NAME (#PCDATA)>
 <!ELEMENT INTERVAL_DATA_ERROR (#PCDATA)>
 <!ELEMENT RSID_VALUES (RSID_STEM*, RSID_VALUE*)>
 <!ELEMENT RSID_STEM (RSID_VALUE+)>
 <ATTLIST RSID_STEM NAME CDATA #REQUIRED>
 <!ELEMENT RSID_VALUE EMPTY>
 <ATTLIST RSID_VALUE
 NAMECDATA #REQUIRED
 TYPE(F|I|S|D) "S"
 VALUECDATA #REQUIRED>
 <!ELEMENT NAME (#PCDATA)>
 <!ELEMENT TYPE (#PCDATA)>
 <!ELEMENT VALUE (#PCDATA)>
]>
```

## XML Example - xmlRulesSchedule

```
<LODESTAR_RUNSCHEDULE>
 <DATASOURCE>
 <NAME>DB1</NAME>
 <CONNECTSTRING>DSN=local; UID=User1; PWD=password ;</CONNECTSTRING>
 <QUALIFIER>TEST</QUALIFIER>
 </DATASOURCE>
 <TIMEOUT>3600</TIMEOUT>
 <USERID>SuperUser</USERID>
 <CONFIG_FILE_NAME> C:\Lodestar\Bin\Lodestar.cfg</CONFIG_FILE_NAME>
 <SCHEDULE_ARGUMENTS >
 <INTERVAL_DATABASE_NAME>RDB </INTERVAL_DATABASE_NAME>
 <RATE_FORM_IDENTIFIER> OPC01:JUR1:RS1:0 </RATE_FORM_IDENTIFIER>
 <START_DATE> 2000/01/01 </START_DATE>
 <STOP_DATE> 2000/01/31 </STOP_DATE>
 <SAVE_RESULTS/>
 <ACCOUNT><ALL/></ACCOUNT>
 <INTERVAL_DATA_ERROR> 1 </INTERVAL_DATA_ERROR>
 <RSID_VALUES>
 <RSID_VALUE NAME="TEST1" TYPE="I" VALUE ="20">
 </ RSID_VALUES >
 </SCHEDULE_ARGUMENTS >
</LODESTAR_RUNSCHEDULE>
```

## Element Descriptions - xmlRulesSchedule

Each of the data elements used in the xmlRulesSchedule argument is described below. The corresponding RUNRS arguments for each element are listed in parentheses.

**LODESTAR\_RUNSCHEDULE:** Root element containing a DATASOURCE and SCHEDULE\_ARGUMENTS element that define the rate schedule to be run.

Elements:

**DATASOURCE:** Element containing connection information for the data source to be accessed by the rate schedule.

Elements:

**NAME:** Name of the data source.

**CONNECTSTRING:** Connection string to the data source, using the following syntax:  
DSN=local; UID=User1; PWD=password;

**QUALIFIER:** Qualifier for the data source.

**TIMEOUT:** Number of seconds after which the process will time out.

**USERID:** User name of the user initiating the rate schedule.

**CONFIG\_FILE\_NAME** (-f): The name of the configuration file that defines the working environment of the Energy Information Platform software (e.g., directs the software where to find and place the application data files and so on). If you do not supply a value for *configfilename*, the system uses the default, which is LODESTAR.CFG. For information about the contents of this configuration file, please refer to the *Oracle Utilities Energy Information Platform Configuration Guide*.

**SCHEDULE\_ARGUMENTS:** Element containing specific arguments and parameters used by the rate schedule.

Elements:

**INTERVAL\_DATABASE\_NAME** (-b): The name of the Interval Database (btrieve file) that contains the interval data for the calculations. If you do not specify a value, the program applies the option associated with the userID. (You can find this value by opening Data Manager or other application program, then select **Tools->Options->Interval Data Source**. It is the value under **Database/Input File Name**.)

**RATE\_FORM\_IDENTIFIER** (-v): The identifier for the rate form. **Required.** OPCO:JUR:RS[VER] are the operating company code, jurisdiction code, rate form code, and optional version number that together identify the rate form. If you do not supply a version number, the program automatically uses the rate form that was applicable on the stop date of the analysis period (see STOP\_DATE). If you do supply a version number, the program uses the trial version indicated (typically used only for testing).

**START\_DATE** (-s): The start date of the date range for the calculations. You can specify the date as an absolute or relative value, as described in **Chapter 8: Setting Up Processing to run in Batch Mode** in the *Oracle Utilities Energy Information Platform Configuration Guide*.

**STOP\_DATE** (-t): The stop date of the date range for the calculations. You can specify the date as an absolute or relative value, as described in **Chapter 8: Setting Up Processing to run in Batch Mode** in the *Oracle Utilities Energy Information Platform Configuration Guide*.

**SAVE\_RESULTS** (-k): Tells the program to save the results, as specified in the rate form. If you leave this switch out of the command, the program ignores all SAVE statements in the rate form (this is useful if you want to debug the rate form before using it in a production mode, for example). If you supply the switch, the program saves interval data, table saves, and CIS records. Since RUNRS does not input an account, there is no saving of determinants to a Bill History record, unless -ai is specified (see below).

**ACCOUNT** (-a): Optional Account All/ID/List name - default is no account. If you specify ID you can save to the Bill History Table.

**CUSTOMER** (-a): Optional Customer All/ID/List name - default is no customer.

Elements: (apply to either ACCOUNT or CUSTOMER)

**ALL:** Specifies whether or not to process all Accounts/Customers. See ACCOUNT/CUSTOMER above.

**ID:** Specifies the Account/Customer ID to process. See ACCOUNT/CUSTOMER above.

**LIST\_NAME:** Specifies the list name of Accounts/Customers to process. See ACCOUNT/CUSTOMER above.

**INTERVAL\_DATA\_ERROR** (-i): Sets interval data handling. Uses the same values as the INTD\_ERROR\_STOP configuration parameter (0,1,2,3). Default # is 1 (error stop). If not set, the **Tools->Options->Error Handling->On Rules Language Interval Data Error** option is used.

**RSID\_VALUES** (-x): Element containing a block of elements defining identifier values to be used for this rate schedule.

Elements:

**RSID\_VALUE** : An individual identifier value to be used for this rate schedule.

Attributes:

**NAME:** The name of the identifier value.

**TYPE:** The type of the identifier value.

**VALUE:** The value for the identifier.

## xmlAnalysis

### Schema - xmlAnalysis

```

<?xml version="1.0" ?>
<xs:schema id="NewDataSet" targetNamespace="http://tempuri.org/Analysis.xsd"
 xmlns:mstns="http://tempuri.org/Analysis.xsd"
 xmlns="http://tempuri.org/Analysis.xsd"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
 attributeFormDefault="qualified" elementFormDefault="qualified">
 <xs:element name="LS_ANALYSIS">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="CONFIG_FILE_NAME" type="xs:string" minOccurs="0" />
 <xs:element name="DATASOURCE" minOccurs="1" maxOccurs="1">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="NAME" type="xs:string" minOccurs="1"
maxOccurs="1" />
 <xs:element name="CONNECTSTRING" type="xs:string" minOccurs="1"
maxOccurs="1" />
 <xs:element name="QUALIFIER" type="xs:string" minOccurs="0"
maxOccurs="1" />
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 <xs:element name="ANALYSIS" minOccurs="0" maxOccurs="1">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="ACCOUNT" minOccurs="0" maxOccurs="1">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="ID" type="xs:string" minOccurs="0"/>
 <xs:element name="LIST" type="xs:string" minOccurs="0"/>
 <xs:element name="ALL" type="xs:boolean" minOccurs="0"/>
 <xs:element name="FILENAME" type="xs:string" minOccurs="0"/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 <xs:element name="CUSTOMER" minOccurs="0" maxOccurs="1">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="ID" type="xs:string" minOccurs="0"/>
 <xs:element name="LIST" type="xs:string" minOccurs="0"/>
 <xs:element name="ALL" type="xs:boolean" minOccurs="0"/>
 <xs:element name="FILENAME" type="xs:string" minOccurs="0"/>
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 <xs:element name="USER_SPECIFIED_STOP" type="xs:string"
minOccurs="0" />
 <xs:element name="NO_APPROVE_ACCOUNTS" type="xs:boolean"
minOccurs="0" />
 <xs:element name="NO_RELATED_SUMMARY_ACCOUNTS" type="xs:boolean"
minOccurs="0" />
 <xs:element name="APPROVE_ACCOUNT_PAGE" type="xs:boolean"
minOccurs="0" />
 <xs:element name="FINAL_BILL" type="xs:boolean" minOccurs="0" />
 <xs:element name="CIS_RECORDS" type="xs:string" minOccurs="0" />
 <xs:element name="TRANSACTION_REPORTS" type="xs:string"
minOccurs="0" />
 <xs:element name="LOGGING_CONFIG_FILE" type="xs:string"
minOccurs="0" />
 <xs:element name="LGNA_TABLES" type="xs:boolean" minOccurs="0" />
 <xs:element name="BILL_PERIODS_TO_CORRECT" type="xs:boolean"
minOccurs="0" />
 </xs:sequence>
 </xs:complexType>
 </xs:element>
 </xs:sequence>
 </xs:complexType>
 </xs:element>

```

```
 <xs:element name="BILL_TYPE" type="xs:string" minOccurs="0" />
 <xs:element name="NO_BH_DATES_TIMES" type="xs:boolean"
minOccurs="0" />
 <xs:element name="NO_BH_DET_VALUES" type="xs:boolean"
minOccurs="0" />
 <xs:element name="REBILL_REASON_CODE" type="xs:string"
minOccurs="0" />
 </xs:sequence>
 <xs:attribute name="TYPE" type="xs:string" />
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>
```

## XML Examples - xmlAnalysis

### Current/Final Bill

```
<?xml version="1.0" encoding="utf-8" ?>
<LS_ANALYSIS>
 <DATASOURCE>
 <NAME>abc</NAME>
 <CONNECTSTRING>DSN=dev;UID=ABC1;PWD=pwd</CONNECTSTRING>
 <QUALIFIER>QUAL1</QUALIFIER>
 </DATASOURCE>
 <CONFIG_FILE_NAME>c:\lodestar\cfg\lodestar.cfg</CONFIG_FILE_NAME>
 <ANALYSIS TYPE="CR">
 <ACCOUNT>
 <LIST>AccountList1</LIST>
 </ACCOUNT>
 <USER_SPECIFIED_STOP>01/01/1999 00:00:00</USER_SPECIFIED_STOP>
 <FINAL_BILL/>
 </ANALYSIS>
</LS_ANALYSIS>
```

### Cancel/ReBill

```
<?xml version="1.0" encoding="utf-8" ?>
<LS_ANALYSIS>
 <DATASOURCE>
 <NAME>abc</NAME>
 <CONNECTSTRING>DSN=dev;UID=ABC1;PWD=pwd</CONNECTSTRING>
 <QUALIFIER>QUAL1</QUALIFIER>
 </DATASOURCE>
 <CONFIG_FILE_NAME>c:\lodestar\cfg\lodestar.cfg</CONFIG_FILE_NAME>
 <ANALYSIS TYPE="BC">
 <CUSTOMER>
 <ID>CUSTOMER1</ID>
 </CUSTOMER>
 <BILL_PERIODS_TO_CORRECT>2</BILL_PERIODS_TO_CORRECT>
 <BILL_TYPE>CR</BILL_TYPE>
 </ANALYSIS>
</LS_ANALYSIS>
```

## Element Descriptions - xmlAnalysis

Each of the data elements used in the xmlAnalysis argument is described below. The corresponding CURFINAL.EXE and BILLCORR.EXE arguments for each element are listed in parentheses.

**LS\_ANALYSIS:** Root element containing a DATASOURCE and ANALYSIS element that define the billing process to be run.

Elements:

**DATASOURCE:** Element containing connection information for the data source to be accessed by the billing process.

Elements:

**NAME:** Name of the data source.

**CONNECTSTRING:** Connection string to the data source, using the following syntax:  
DSN=local; UID=User1; PWD=password;

**QUALIFIER:** Qualifier for the data source.

**CONFIG\_FILE\_NAME** (-f): The name of the configuration file that defines the working environment of the Energy Information Platform software (e.g., directs the software where to find and place the application data files and so on). If you do not supply a value for *configfilename*, the system uses the default, which is LODESTAR.CFG. For information about the contents of this configuration file, please refer to the *Oracle Utilities Energy Information Platform Configuration Guide*.

**ANALYSIS:** Element containing specific arguments and parameters used by the billing process.

Attributes:

**TYPE:** Specifies the type of analysis. Can be either BC (Bill Correction), or CF (Current/Final bill).

Elements:

**ACCOUNT** (-a): Account All/ID/List name/File name. The default is all accounts.

**CUSTOMER** (-a): Customer All/ID/List name/File name. The default is all customers.

Elements: (apply to either ACCOUNT or CUSTOMER)

**ID:** Specifies the Account/Customer ID to process. See ACCOUNT/CUSTOMER above.

**LIST:** Specifies the list name of Accounts/Customers to process. See ACCOUNT/CUSTOMER above.

**ALL:** Specifies whether or not to process all Accounts/Customers. See ACCOUNT/CUSTOMER above.

**FILENAME:** Specifies the path and file name of a file containing Accounts/Customer IDs to process. See ACCOUNT/CUSTOMER above.

**USER\_SPECIFIED\_STOP** (-t, CURFINAL): The User Specified Stop Date, in MM/DD/YYYY format. The default is computed.

**NO\_APPROVE\_ACCOUNTS** (-s): If you include this element, accounts are not approved. The default is to Approve.

**NO\_RELATED\_SUMMARY\_ACCOUNTS** (-m): If you include this element, related summary accounts are not processed. The default is to process related accounts.

**FINAL\_BILL** (-n, CURFINAL): Indicates that the bill is to be a final bill.

**CIS\_RECORDS** (-o): The name of the output file used by the CIS system. The default is LODESTAR.CIS.

**TRANSACTION\_REPORTS** (-w): The filename for transaction reports [default: YYYYMMDD.HHMM].

**LOGGING\_CONFIG\_FILE** (-lcfg): Name of an optional logging configuration file that specifies where error and log messages are sent.

**LGNA\_TABLES** (-o): Indicates that output should be sent to the LGNA tables.

**BILL\_TYPE** (-t, BILLCORR): The Bill type: Cancel/Rebill (CR), Adjustment (A), Cancel (C), Rebill (R). The default is Cancel/Rebill (CR).

**BILL\_PERIODS\_TO\_CORRECT** (-b): The number of bill periods to correct. The default is 0.

**NO\_BH\_DATES\_TIMES** (-h): If you include this element, Bill History dates and times will not be used. The default is to use them. **Note:** This element only applies if dates/times can be computed from interval data.

**NO\_BH\_DET\_VALUES** (-v): If you include this element, current Bill History determinant values will not be used. The default is to use them. **Note:** This element only applies if dates/times can be computed from interval data.

**REBILL\_REASON\_CODE** (-d): The reason for the bill correction.

**APPROVE\_ACCOUNT\_PAGE** (-1): If you include this element, each Account is approved if no error occurs. The default is to approve each group[ of accounts.

# Return Values

## xmlSaveToXML

### DTD - xmlSaveToXML

```
<!DOCTYPE RSID_VALUES
[
<!ELEMENT RSID_VALUES (RSID_STEM*, RSID_VALUE*)>
<!ELEMENT RSID_STEM (RSID_VALUE+)>
<ATTLIST RSID_STEM NAME CDATA #REQUIRED>
<!ELEMENT RSID_VALUE EMPTY>
<ATTLIST RSID_VALUE
 NAMECDATA#REQUIRED
 TYPE(F|I|S|D)"S"
 VALUECDATA#REQUIRED>
]>
```

### XML Example - xmlSaveToXML

```
<RSID_VALUES>
 <RSID_STEM NAME="VAL1">
 <RSID_VALUE NAME="UOM" TYPE="S" VALUE="79"/>
 <RSID_VALUE NAME="D_KWH" TYPE="F" VALUE="1000"/>
 </RSID_STEM>
 <RSID_VALUE NAME="MY_RS_ID" TYPE="S" VALUE="TEST"/>
 <RSID_VALUE NAME="MY_RS_ID2" TYPE="F" VALUE="20.99"/>
</RSID_VALUES>
```

### Element Descriptions - xmlSaveToXML

The use of each individual attribute and element in the xmlLodestarRunScheduleResult argument is described below.

**RSID\_VALUES:** Element containing a block of elements defining identifier values returned from the rate schedule.

Elements:

**RSID\_STEM :** Element specifying a stem identifier and containing one or more RSID\_VALUE elements, each specifying a tail identifier.

Attributes:

**NAME:** The name of the stem identifier.

**RSID\_VALUE :** An individual identifier value return from the rate schedule. When found within a RSID\_STEM element, RSID\_VALUE elements represent tail identifiers.

Attributes:

**NAME:** The name of the identifier value.

**TYPE:** The type of the identifier value.

**VALUE:** The value for the identifier.

## xmlAnalysisResult

### Schema - xmlAnalysisResult

```
<?xml version="1.0"?>
<xs:schema targetNamespace="http://tempuri.org/~vs1DC.xsd"
 xmlns:mstns="http://tempuri.org/~vs1DC.xsd"
 xmlns="http://tempuri.org/~vs1DC.xsd"
 xmlns:xs="http://www.w3.org/2001/XMLSchema"
 xmlns:msdata="urn:schemas-microsoft-com:xml-msdata"
 attributeFormDefault="qualified" elementFormDefault="qualified">
 <xs:element name="LS_ANALYSIS">
 <xs:complexType>
 <xs:sequence>
 <xs:element name="RESULT" minOccurs="0" maxOccurs="1">
 <xs:complexType>
 <xs:attribute name="CODE" type="xs:string" />
 </xs:complexType>
 </xs:element>
 </xs:sequence>
 <xs:attribute name="TYPE" type="xs:string" />
 </xs:complexType>
 </xs:element>
</xs:schema>
```

### XML Example - xmlAnalysisResult

```
<?xml version="1.0" encoding="utf-8"?>
<LS_ANALYSIS TYPE="BC">
 <RESULT CODE="99" />
</LS_ANALYSIS>
```

### Element Descriptions - xmlAnalysisResult

The use of each individual attribute and element in the xmlAnalysisResult argument is described below.

**LS\_ANALYSIS:** Element containing the result of the billing process.

Attributes:

**TYPE:** Specifies the type of billing process. Can be either BC (Bill Correction), or CF (Current/Final bill).

Elements:

**RESULT :** Element containing result code for billing process.

Attributes:

**CODE:** The result code of the billing process. Can be one of the following:

- 0: All accounts that could be billed were billed successfully.
- 1: Analysis could not start.
- 99: At least one account failed to bill.

# Chapter 19

---

## Energy Information Platform Work Queues Interface

This chapter describes the methods/functions available to external systems through the Energy Information Platform Work Queues interface (IWorkQueue). These methods allow users to perform a number of work queue functions, including:

- Open
- Assign
- Unassign
- Update
- Resolve
- Approve
- Reject
- Close
- Reopen
- Exists

# Methods, Interfaces, and Syntax

The methods, interface objects, and syntax for the Work Queues interface are as follows:

## Open

**Description:** Used to open a work queue item.

**Method Name:** Open

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Open([in] BSTR xmlDataSource,
 [in] BSTR xmlWQItem,
 [out, retval] BSTR* xmlWQItemOut);
```

## Assign

**Description:** Used to assign a work queue item to a specified user.

**Method Name:** Assign

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Assign([in] BSTR xmlDataSource,
 [in] BSTR xmlWQItem,
 [in] BSTR strUserId,
 [in, optional] BSTR strNote);
```

## Unassign

**Description:** Used to unassigned a previously assigned work queue item.

**Method Name:** Unassign

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Unassign([in] BSTR xmlDataSource,
 [in] BSTR xmlWQItem,
 [in] BSTR strUserId,
 [in, optional] BSTR strNote);
```

## Update

**Description:** Used to update a specified attribute of an open work queue item.

**Method Name:** Update

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Update([in] BSTR xmlDataSource,
 [in] BSTR xmlWQItem,
 [in] BSTR strUserId,
 [in, optional] BSTR strNote);
```

**Note:** only Queue Code, Priority Level, WorkBytime or Data elements can be updated.

## Resolve

**Description:** Used to resolve an open work queue item.

**Method Name:** Resolve

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Resolve([in] BSTR xmlDataSource,
 [in] BSTR xmlWQItem,
 [in] BSTR strUserId,
 [in, optional] BSTR strNote);
```

## Approve

**Description:** Used to approve an open work queue item.

**Method Name:** Approve

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Approve([in] BSTR xmlDataSource,
 [in] BSTR xmlWQItem,
 [in] BSTR strUserId,
 [in, optional] BSTR strNote);
```

## Reject

**Description:** Used to reject an open work queue item.

**Method Name:** Reject

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Reject([in] BSTR xmlDataSource,
```

```
[in] BSTR xmlWQItem,
[in] BSTR strUserId,
[in, optional] BSTR strNote);
```

## Close

**Description:** Used to close an open work queue item.

**Method Name:** Close

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Close([in] BSTR xmlDataSource,
 [in] BSTR xmlWQItem,
 [in] BSTR strUserId,
 [in, optional] BSTR strNote);
```

## Reopen

**Description:** Used to reopen a previously closed work queue item.

**Method Name:** Reopen

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Reopen([in] BSTR xmlDataSource,
 [in] BSTR xmlWQItem,
 [in] BSTR strUserId,
 [in, optional] BSTR strNote);
```

## Exists

**Description:** Used to determine if a specified work queue item exists.

**Method Name:** Exists

**Interface:** IWorkQueue

**DLL Name:** LSWorkQueue.DLL

**Program ID:** LSWorkQueue.WorkQueue

**Syntax:**

```
HRESULT Exists([in] BSTR xmlDataSource,
 [in] BSTR xmlWQItem,
 [out, retval] BOOL bExists);
```

## Interface Arguments

The methods available in the Work Queues interface use the following arguments:

### **xmlDataSource Argument**

The xmlDataSource argument is an xml string that contains database connection and other related information. A DTD, xml example, and data element descriptions for this argument can be found on page 15-7.

### **xmlWQItem Argument**

The xmlWQItem argument is an xml string that contains a work queue item. Xml examples and data element descriptions for this argument can be found on page 19-6.

### **xmlWQItemOut Argument**

The xmlWQItemOut argument is an xml string that contains a work queue item. An xml example and data element descriptions for this argument can be found on page 19-13.

### **strUserId Argument**

The strUserID argument is a string that contains the user id of the user invoking the method.

### **strNote Argument**

The strNote argument is a string that contains an optional note related to the work queue item.

### **bExists Argument**

The bExists argument is a Boolean value that indicates if a specified work queue item exists or not. A value of 1 indicates the item exists, a value of 0 (zero) indicate the item does not exist. This argument is returned from the Exists method only.

## Input Values

Xml examples and data element descriptions of the xmlWQItem argument used as an input value for the Work Queues interface (IWorkQueue) are provided below.

### xmlWQItem XML Elements

```
<WQITEM>
 <UID/>
 <TYPE/>
 <QUEUE/>
 <PRIORITYLEVEL/>
 <WORKBYTIME/>
 <OPENEDTIME/>
 <OPENEDBYUSERID/>
 <OPENEDNOTE/>
 <ASSIGNEDTOUSERID/>
 <ASSIGNEDTOTIME/>
 <APPRSTEPNUM/>
 <APPRSTEPCOUNT/>
 <APPRLEVEL/>
 <APPROVALONLY/>
 <APPROVED/>
 <PRODUCT/>
 <PROCESSNAME/>
 <PROCESSID/>
 <UIDRATEFORM/>
 <READDATE/>
 <ACCOUNTID/>
 <AMOUNT/>
 <ITEMDATA/>
 <CHANNELNO/>
 <INTDSTARTTIME/>
 <JURISCODE/>
 <METERID/>
 <OPCOCODE/>
 <RECORDERID/>
 <TRANSACTIONNO/>
</WQITEM>
```

### XML Examples - xmlWQItem

#### Open

```
<WQITEM>
 <UID/>
 <TYPE/>
 <TYPE VALTYPE="NEW">REGULAR</TYPE>
 <QUEUE/>
 <PRIORITYLEVEL/>
 <WORKBYTIME/>
 <OPENEDBYUSERID>Karen</OPENEDBYUSERID>
 <OPENEDNOTE>This item is not "Approval Only"</OPENEDNOTE>
 <ASSIGNEDTOUSERID/>
 <ASSIGNEDTOTIME/>
 <APPRSTEPNUM/>
 <APPRSTEPCOUNT/>
 <APPROVALONLY/>
 <ACCOUNTID>8000029</ACCOUNTID>
 <AMOUNT>100000.00</AMOUNT>
 <CHANNELNUM>1</CHANNELNUM>
 <INTDSTARTTIME>2002-12-06T15:08:10</INTDSTARTTIME>
 <ITEMDATA>Warning: bill Exceeds $100000.00</ITEMDATA>
 <JURISCODE>SF</JURISCODE>
 <METERID">25</METERID>
 <OPCOCODE/>
```

```

<OPCOCODE>GECO</OPCOCODE>
<PROCESSID>Steven</PROCESSID>
<READDATE>2002-12-10</READDATE>
<RECORDERID>2002</RECORDERID>
<TRANSACTIONNO>4561</TRANSACTIONNO>
<UIDRATEFORM>444</UIDRATEFORM>
</WQITEM>

```

## Assign

For assigning an item to a new user:

```

<WQITEM>
 <UID>160</UID>
 <TYPE> Code_1</TYPE>
 <QUEUE> Queue_1</QUEUE>
 <PRIORITYLEVEL>1</PRIORITYLEVEL>
 <WORKBYTIME>2002-11-25T12:44:46</WORKBYTIME>
 <ASSIGNEDTOUSERID/>
 <ASSIGNEDTOTIME/>
 <APPRSTEPNUM>1</APPRSTEPNUM>
 <APPRSTEPCOUNT>0</APPRSTEPCOUNT>
 <APPRLEVEL>2</APPRLEVEL>
 <ASSIGNEDTOUSERID VALTYPE="NEW">Steven</ASSIGNEDTOUSERID>
 <APPROVALONLY>FALSE</APPROVALONLY>
</WQITEM>

```

For reassigning an item from a previous user to a new user:

```

<WQITEM>
 <UID>160</UID>
 <TYPE> Code_1</TYPE>
 <QUEUE> Queue_1</QUEUE>
 <PRIORITYLEVEL>1</PRIORITYLEVEL>
 <WORKBYTIME>2002-11-25T12:44:46</WORKBYTIME>
 <ASSIGNEDTOUSERID>Karen</ASSIGNEDTOUSERID>
 <ASSIGNEDTOTIME>2002-11-25T12:00:00</ASSIGNEDTOTIME>
 <APPRSTEPNUM>1</APPRSTEPNUM>
 <APPRSTEPCOUNT>0</APPRSTEPCOUNT>
 <APPRLEVEL>2</APPRLEVEL>
 <ASSIGNEDTOUSERID VALTYPE="NEW">Steven</ASSIGNEDTOUSERID>
 <APPROVALONLY>FALSE</APPROVALONLY>
</WQITEM>

```

## Unassign

```

<WQITEM>
 <UID>20</UID>
 <TYPE> Code_1</TYPE>
 <QUEUE>TORINO_WQ</QUEUE>
 <PRIORITYLEVEL>1</PRIORITYLEVEL>
 <WORKBYTIME>2002-11-07T16:53:02</WORKBYTIME>
 <ASSIGNEDTOUSERID>karen</ASSIGNEDTOUSERID>
 <ASSIGNEDTOTIME>2002-11-07T14:54:46</ASSIGNEDTOTIME>
 <APPRSTEPNUM>1</APPRSTEPNUM>
 <APPRSTEPCOUNT>0</APPRSTEPCOUNT>
 <APPRLEVEL>2</APPRLEVEL>
 <ASSIGNEDTOUSERID VALTYPE="NEW" />
 <APPROVALONLY>FALSE</APPROVALONLY>
</WQITEM>

```

## Update

```

<WQITEM>
 <UID>142</UID>
 <TYPE>Code_1</TYPE>
 <QUEUE>Queue_1</QUEUE>
 <PRIORITYLEVEL>1</PRIORITYLEVEL>
 <WORKBYTIME>2002-11-22T12:53:14</WORKBYTIME>

```

```
<ASSIGNEDTOUSERID/>
<ASSIGNEDTOTIME/>
<APPRSTEPNUM/>
<APPRSTEPCOUNT/>
<QUEUE VALTYPE="NEW"> TORINO_WQ </QUEUE>
<PRIORITYLEVEL VALTYPE="NEW">2</PRIORITYLEVEL>
<WORKBYTIME VALTYPE="NEW">2002-11-22T11:54:00</WORKBYTIME>
</WQITEM>
```

## Resolve

```
<WQITEM>
<UID>160</UID>
<TYPE> Code_1</TYPE>
<QUEUE> Queue_1</QUEUE>
<PRIORITYLEVEL>1</PRIORITYLEVEL>
<WORKBYTIME>2002-11-25T12:44:46</WORKBYTIME>
<ASSIGNEDTOUSERID/>
<ASSIGNEDTOTIME/>
<APPRSTEPNUM/>
<APPRSTEPCOUNT/>
<APPRLEVEL/>
<APPROVALONLY/>
</WQITEM>
```

## Approve

```
<WQITEM>
<UID>62</UID>
<TYPE> Code_1</TYPE>
<QUEUE> Queue_1</QUEUE>
<PRIORITYLEVEL>2</PRIORITYLEVEL>
<WORKBYTIME>2002-11-19T10:19:07</WORKBYTIME>
<ASSIGNEDTOUSERID>Steven</ASSIGNEDTOUSERID>
<ASSIGNEDTOTIME>2002-11-18T10:19:07</ASSIGNEDTOTIME>
<APPRSTEPNUM>1</APPRSTEPNUM>
<APPRSTEPCOUNT>0</APPRSTEPCOUNT>
<APPRLEVEL>Level_Code_1</APPRLEVEL>
<APPROVALONLY/>
</WQITEM>
```

## Reject

```
<WQITEM>
<UID>160</UID>
<TYPE> Code_1</TYPE>
<QUEUE> Queue_1</QUEUE>
<PRIORITYLEVEL>1</PRIORITYLEVEL>
<WORKBYTIME>2002-11-19T10:19:07</WORKBYTIME>
<ASSIGNEDTOUSERID>Steven</ASSIGNEDTOUSERID>
<ASSIGNEDTOTIME>2002-11-18T10:19:07</ASSIGNEDTOTIME>
<APPRSTEPNUM>1</APPRSTEPNUM>
<APPRSTEPCOUNT>0</APPRSTEPCOUNT>
<APPRLEVEL>Level_Code_1</APPRLEVEL>
<APPROVALONLY/>
</WQITEM>
```

## Close

```
<WQITEM>
<UID>52</UID>
<TYPE> Code_1</TYPE>
<QUEUE> Queue_1</QUEUE>
<PRIORITYLEVEL>1</PRIORITYLEVEL>
<WORKBYTIME>2002-11-14T00:00:00</WORKBYTIME>
<OPENEDTIME>2002-11-13T13:31:16</OPENEDTIME>
<ASSIGNEDTOUSERID/>
<ASSIGNEDTOTIME/>
<APPRSTEPNUM/>
```

```
<APPRSTEPCOUNT/>
<APPRLEVEL/>
<APPROVALONLY/>
</WQITEM>
```

## Reopen

```
<WQITEM>
<UID>50</UID>
<TYPE> Code_1</TYPE>
<QUEUE> Queue_1</QUEUE>
<PRIORITYLEVEL>1</PRIORITYLEVEL>
<ASSIGNEDTOUSERID/>
<ASSIGNEDTOTIME/>
<APPRSTEPNUM/>
<APPRSTEPCOUNT/>
<APPRLEVEL/>
<APPROVALONLY/>
</WQITEM>
```

## Exists

```
<WQITEM>
<UID>161</UID>
<TYPE> Code_1</TYPE>
<QUEUE> Queue_1</QUEUE>
<PRIORITYLEVEL>1</PRIORITYLEVEL>
<WORKBYTIME>2002-11-27T10:57:36</WORKBYTIME>
<ASSIGNEDTOUSERID/>
<ASSIGNEDTOTIME/>
<APPRSTEPNUM>1</APPRSTEPNUM>
<APPRSTEPCOUNT>1</APPRSTEPCOUNT>
<APPRLEVEL>Level_Code_1</APPRLEVEL>
<ASSIGNEDTOUSERID/>
<APPROVALONLY/>
</WQITEM>
```

## Element Descriptions - xmlWQItem

Each of the data elements used by the xmlWQItem argument is described below.

Elements:

**UID:** A unique identifier for the work queue item.

**TYPE:** The work queue type for the item. This is the only attribute that is required when opening a new work queue item. Supports the VALTYPE="NEW" attribute for new work queue items.

**QUEUE:** The work queue for the item. If not explicitly provided then the default queue of the corresponding type will be used. Supports the VALTYPE="NEW" attribute when updating work queue items.

**PRIORITYLEVEL:** The priority level of the work queue item. If not explicitly provided then the default priority level of the corresponding type will be used. Supports the VALTYPE="NEW" attribute when updating work queue items.

**WORKBYTIME:** (Open items only) This is the optional time by which the item is expected to be resolved. If not provided then it is calculated from the Default Work By Hours values of the corresponding type. Supports the VALTYPE="NEW" attribute when updating work queue items.

**OPENEDTIME:** The time when the item was opened. This is automatically determined when the item is opened and can never be subsequently changed.

**OPENEDNOTE:** An optional note for the item, created when the item was opened.

**ASSIGNEDTOUSERID:** (Opened items only) UserId of user to which the item is currently assigned. This will change each time the item is unassigned, reassigned, resolved, approved, rejected, or closed. Requires the VALTYPE="NEW" attribute when updating work queue items.

**ASSIGNEDTOTIME:** (Opened items only) Time at which item was assigned to current user. This will also change each time the item is unassigned, reassigned, resolved, approved, rejected, or closed. Requires the VALTYPE="NEW" attribute when updating work queue items.

**APPRSTEPNUM:** (Opened items only) Current approval step of corresponding approval process that item is on. This will change as the item move through the approval process.

**APPRSTEPCOUNT:** (Opened items only) Current number of approvals for this approval step.

**APPRLEVEL:** (Opened items only) Approval level corresponding to current approval step described above.

**APPROVALONLY:** Provided by corresponding type. If true, indicates that the item requires a final approved/rejected state.

**APPROVED:** Final approved/rejected state for approval only items.

**PRODUCT:** Optional product associated with the item - used for filtering (i.e. Oracle Utilities Billing Component). If not explicitly provided then the default product of the corresponding type will be used, else it is NULL. If provided, the PRODUCT must have a corresponding record in the LODESTAR Product table.

**PROCESSNAME:** Optional process name associated with the item - used for filtering (i.e. AutoBill). If provided, the PRODUCTNAME must have a corresponding record in the Business Process table.

**PROCESSID:** Optional identifying information that can be used to distinguish one instance of a process from another.

**UIDRATEFORMVERSION:** Optional rate form version from which item was generated.

**READDATE:** Optional read date of current billing cycle from which item was generated. Will be used together with the Stop Billing flag similar to how ACCOUNTNOTES are used.

**UIDACCOUNT:** Optional. Unique identifier for the account associated with the work queue item.

**ACCOUNTID:** Optional. Account ID for the account associated with the work queue item.

**AMOUNT:** Optional. An amount associated with the work queue item.

**DATA:** Optional additional data specific to the item, represented as an XML structure. Supports the VALTYPE="NEW" attribute when updating work queue items.

## Specifying New Values

When updating work queue items via the Update, Assign, or Unassign methods, elements for values to be updated should be listed twice, first with the current value, followed by the new value. The element containing the new value should also include the VALTYPE attribute (with a value of "NEW").

For updating an item: In the example below, Priority Level is changed from 1 to 2, and the Work By Time is changed from 2002-11-22T12:53:14 to 2002-11-22T11:54:00.

```
<WQITEM>
 <UID>142</UID>
 <TYPE>Code_1</TYPE>
 <QUEUE>Queue_1</QUEUE>
 <PRIORITYLEVEL>1</PRIORITYLEVEL>
 <WORKBYTIME>2002-11-22T12:53:14</WORKBYTIME>
 <ASSIGNEDTOUSERID/>
 <ASSIGNEDTOTIME/>
 <APPRSTEPNUM/>
 <APPRSTEPCOUNT/>
 <QUEUE VALTYPE="NEW"> TORINO_WQ </QUEUE>
 <PRIORITYLEVEL VALTYPE="NEW">2</PRIORITYLEVEL>
 <WORKBYTIME VALTYPE="NEW">2002-11-22T11:54:00</WORKBYTIME>
</WQITEM>
```

For assigning an item to a new user:

```
<WQITEM>
 <UID>160</UID>
 <TYPE> Code_1</TYPE>
 <QUEUE> Queue_1</QUEUE>
 <PRIORITYLEVEL>1</PRIORITYLEVEL>
 <WORKBYTIME>2002-11-25T12:44:46</WORKBYTIME>
 <ASSIGNEDTOUSERID/>
 <ASSIGNEDTOTIME/>
 <APPRSTEPNUM>1</APPRSTEPNUM>
 <APPRSTEPCOUNT>0</APPRSTEPCOUNT>
 <APPRLEVEL>2</APPRLEVEL>
 <ASSIGNEDTOUSERID VALTYPE="NEW">Steven</ASSIGNEDTOUSERID>
 <APPROVALONLY>FALSE</APPROVALONLY>
</WQITEM>
```

For reassigning an item from a previous user to a new user:

```
<WQITEM>
 <UID>160</UID>
 <TYPE> Code_1</TYPE>
 <QUEUE> Queue_1</QUEUE>
 <PRIORITYLEVEL>1</PRIORITYLEVEL>
 <WORKBYTIME>2002-11-25T12:44:46</WORKBYTIME>
 <ASSIGNEDTOUSERID>Karen</ASSIGNEDTOUSERID>
 <ASSIGNEDTOTIME>2002-11-25T12:00:00</ASSIGNEDTOTIME>
 <APPRSTEPNUM>1</APPRSTEPNUM>
 <APPRSTEPCOUNT>0</APPRSTEPCOUNT>
 <APPRLEVEL>2</APPRLEVEL>
 <ASSIGNEDTOUSERID VALTYPE="NEW">Steven</ASSIGNEDTOUSERID>
 <APPROVALONLY>FALSE</APPROVALONLY>
</WQITEM>
```

## Custom Parameter Values

In addition to the elements listed, work queue items may have additional custom parameter values explicitly provided when opened. These items must have corresponding records in the Work Queue Custom Parameter table, and must match the column names in the Work Queue Open Item and Work Queue Closed Item tables in the Oracle Utilities Data Repository.

## Return Values

The data returned from the Work Queues interfaces is described in the following xml example, and data element descriptions.

### xmlWQItemOut

The xmlWQItemOut argument is the same as the xmlWQItem argument.

#### **Element Descriptions - xmlWQItemOut**

The data elements for the xmlWQItemOut argument are the same as the xmlWQItem argument.



# Chapter 20

---

## Energy Information Platform Interval Data Interfaces

This chapter describes the methods/functions available to external systems through the Energy Information Platform Interval Data interfaces (LSINTD). These interfaces allow users to work with interval data stored in the Oracle Utilities Data Repository. This chapter includes:

- **LSINTD Interface Methods, and Parameter Descriptions**
- **Interval Data Source Providers**
- **Sample Application using LSINTD**
- **INTDVB Compatibility**

# LSINTD Interface Methods, and Parameter Descriptions

This section details the interfaces, methods and parameters in the published Energy Information Platform COM interval data interfaces.

## IIntDDataSource Interface

The IIntDDataSource object uses the provided connection string sent to the connect method to connect an interval datasource. A data source is defined as an interval data storage media, and may be RDB, BTE, LSE or XML. The IIntDDataSource object has a number of methods which may be called. Published interfaces appear in bold face and detailed below.

### Interface Definition

```
[
 odl,
 uuid(D4F2C360-E3E3-41f0-AF1F-0708F863D437),
 version(1.0),
 dual,
 oleautomation
]
interface IIntDDataSource : IDispatch {
[id(1), helpstring("method Connect")]
HRESULT Connect(BSTR ConnectString);
[id(2), helpstring("method Flush")]
HRESULT Flush();
[propget, id(3), helpstring("property Recorders")]
HRESULT Recorders([out, retval] IIntDRecorderList* *pVal);
[propget, id(4), helpstring("property Cuts")]
HRESULT Cuts([out, retval] IIntDCutList* *pVal);
[id(5), helpstring("method LoadDates")]
HRESULT LoadDates([in] BSTR Recorder,
 [in] long Channel,
 [in] DATE StartDate,
 [in] DATE StopDate,
 [out, retval] IIntDCut** RetVal);
[id(6), helpstring("method GetCutByKey")]
HRESULT GetCutByKey([in] BSTR Recorder,
 [in] long Channel,
 [in] DATE StartDate,
 [out, retval] IIntDPureCut** RetVal);

[id(7), helpstring("method GetCutByUID")]
HRESULT GetCutByUID([in] BSTR UID, [out, retval] IIntDPureCut** RetVal);

[id(8), helpstring("method GetImporter")]
HRESULT GetImporter([out, retval] IIntDImporter** RetVal);
};
```

This interface has properties and methods that are not published. The published and user-available functions are listed below.

### Published Method: Connect

The IIntDDataSource object uses the provided connection string to determine which data provider must be used, loads it dynamically, and forwards the connection string to it.

The Connect method accepts one argument called "ConnectionString". This argument defines a provider which will be used to connect to an interval data source. This argument also contains parameters specific for the current session which define interval data behavior. The published providers are described later in this document.

The ConnectString argument is defined by the following XML format:

```
<INTDATASOURCE>
 <PROVIDER PROGID="provider's ProgId">
```

```

 . . . provider-specific parameters . . . (such as RDB-DataSource XML or BTE
filename)
 </PROVIDER>
 <SESSION>
 . . . connection-specific parameters . . .
 </SESSION>
</INTDATASOURCE>

```

The **PROGID** attribute of the **PROVIDER** element defines a ProgID of a COM component that implements a particular interval data source provider. This COM component must support the internal IIntDProvider interface.

The **SESSION** element contains session-specific parameters. This element may contain settings for the current session and are provider-independent. This means that they affect generic cut functions and behavior and not data access behavior. The session is defined in detail below. The following XML format is used to define the session:

```

<SESSION>
 [<QualityCode>Character</QualityCode>]
 [<DayStart>Integer</DayStart>]
 [<INTD_NEW_JOIN [INTDMERGESELECT="RECENT|AVERAGE|BEST|MAXIMUM"]
[CHECKVALID="Y|N"] />]
 [<CHECK_UOM_MATCHING [CHECK_COMPATIBILITY="Y|N"] />]
 [<DO_NOT_USE_UOM_FOR_SCALING/>]
</SESSION>

```

The elements used within the SESSION element include:

- **QualityCode:** Ignore interval data values when status code of value is worse than this QualityCode. Valid values for this element are [A-Z] | [0-9], space, or no value. If no value is present then a status code of space is used. If the QualityCode element is not present, the default used is a value of 8. If the status code of an interval is worse than the character specified in the QualityCode code element, then that interval will be returned as a missing interval (Value of 0, Status code '9').
- **DayStart:** This is used as the start of the day. Valid values are 0 - 11, where 0 is for Midnight, 1 is for 1 A.M., 2 is for 2 A.M., up to 11 for 11 A.M. If DayStart element is not present, the default value is 0 (Midnight).
- **INTD\_NEW\_JOIN:** Determines the criteria when merging interval data cuts. See **Interval Data Merge** on page 2-14 of the *Data Manager User's Guide* for more information.
- **CHECK\_UOM\_MATCHING:** If this element exists, the interval data component checks UOM Compatibility when merging cuts across a specified date range; by default the component does not check UOM compatibility. If **CHECK\_COMPATIBILITY** is 'Y' component will check the UOM's compatibility (matching).
- **DO\_NOT\_USE\_UOM\_FOR\_SCALING:** If this element exists, interval data is totalized when scaling or aggregating, regardless of the Totalize Method flag specified in the Unit-of-Measure table.

#### Example 1 (QualityCode only)

```

<SESSION>
 <QualityCode></QualityCode> (a space is assumed)
</SESSION>

```

#### Example 2 (Use defaults)

```

<SESSION/>

```

**Example 3**

```
<SESSION>
 <QualityCode>7</QualityCode>
 <DayStart></DayStart> (This starts the day at 12:00 AM)
 <CHECK_UOM_MATCHING CHECK_COMPATIBILITY="Y"/> (Do not allow merging cuts
with incompatible UOM's)
</SESSION>
```

**Published Method: LoadDates**

This method creates and loads an IIntDCut object from the then-active provider (data store) for a user-specified date range using all settings specified in the Session parameter.

```
HRESULT LoadDates (
 [in] BSTR Recorder,
 [in] BSTR Channel,
 [in] DATE StartDate,
 [in] DATE StopDate,
 [out, retval] IIntDCut** pRetVal);
```

This method accepts 4 arguments and returns a pointer to 1 object.

- **Recorder:** The Recorder BSTR may contain a maximum of 64 bytes.
- **Channel:** The channel BSTR may contain a positive value of less than 32767.
- **StartDate:** StartDate is a Date Value representing the date of the beginning of the range to expect interval data to be returned.
- **StopDate:** StopDate is a Date Value that indicates the end time of the interval data request.
- **pRetVal:** A pointer to the cut object (IIntDCut) created by the method over the specified date range will be returned.

**Published Method: GetCutByKey**

This method loads an IIntDPureCut object from the then-active provider (data store) for a user-specified date range using all settings specified in the Session parameter.

The parameters to this method detail a full cut key which uniquely identifies an Interval Data Cut.

```
[id(6), helpstring("method GetCutByKey")]
HRESULT GetCutByKey([in] BSTR Recorder,
 [in] long Channel,
 [in] DATE StartDate,
 [out, retval] IIntDPureCut** RetVal);
```

This method accepts 3 arguments and returns pointers to 2 objects.

- **Recorder:** The Recorder BSTR may contain a maximum of 64 bytes.
- **Channel:** The channel BSTR may contain a positive value of less than 32767.
- **StartDate:** StartDate is a Date Value representing the date of the requested cut.
- **Property: Recorders:** A pointer to a newly created IIntDRecorderList object is returned to the calling program.
- **Property: Cuts:** A pointer to a newly created IIntdCutList object which contains all cuts is returned to the calling program.

## IIntDRecorderList Interface

The IIntDRecorderList interface provides basic navigation through the list of IIntDRecorder. It extends the COM interface IEnumVARIANT that provides ability to use “for each” statements in Visual Basic and JScript.

### Interface Definition

```
[
 object,
 uuid(77F1D0B4-D5A6-472F-B0AB-855739021B3A),
 dual,
 helpstring("IIntDRecorderList Interface"),
 pointer_default(unique)
]
interface IIntDRecorderList : IDispatch
{
 [propget, id(1), helpstring("property Count")]
 HRESULT Count([out, retval] long *pVal);
 [id(2), helpstring("method Add")]
 HRESULT Add([in] BSTR RecorderId, [out, retval] IIntDRecorder **pVal);
 [propget, id(3), helpstring("property Item")]
 HRESULT Item([in] BSTR RecorderId, [out, retval] IIntDRecorder* *pVal);
 [id(4), helpstring("method Remove")]
 HRESULT Remove(BSTR RecorderId);
 [id(5), helpstring("method RemoveAll")]
 HRESULT RemoveAll();
 [id(0xffffffffc), propget, restricted, hidden, helpstring("property _newEnum")]
 HRESULT _newEnum([out, retval] IUnknown* *pVal);
};
```

**Property: \_newEnum** (Interface IEnumVARIANT): This property is implemented by using rules of the COM environment, and provides “for each” functionality.

**Property: Count:** Contains list length.

**Property: Item:** Returns IIntDRecorder object for the particular RecorderID.

### Method: Add

Inserts the new recorder ID into a recorder list and returns a newly created IIntDRecorder object.

### Method: Remove

Removed recorder from a recorders list.

### Method: RemoveAll

Removes all recorders from a recorders list.

## IIntDRecorder Interface

The IIntDRecorder interface provides access to an IIntDRecorder object.

### Interface Definition

```
[
 object,
 uuid(864AB91F-1B89-44DB-8C91-32BDE4965673),
 dual,
 helpstring("IIntDRecorder Interface"),
 pointer_default(unique)
]
interface IIntDRecorder : IDispatch
{
 [propget, id(1), helpstring("property RecorderId")]
 HRESULT RecorderId([out, retval] BSTR *pVal);
 [propget, id(2), helpstring("property Channels")]
 HRESULT Channels([out, retval] IIntDChannelList* *pVal);
};
```

**Property: RecorderId:** This property contains a recorder ID.

**Property: Channels:** This property returns an IIntDChannelList object.

## IIntDChannelList Interface

The IIntDChannelList interface provides basic navigation through the list of IIntDChannel. It extends the COM interface IEnumVARIANT that provides ability to use “for each” statements in Visual Basic and JScript.

### Interface Definition

```
[
 object,
 uuid(9D6893A1-95A5-4153-BF93-FB2B7CBE23D1),
 dual,
 helpstring("IIntDChannelList Interface"),
 pointer_default(unique)
]
interface IIntDChannelList : IDispatch
{
 [propget, id(1), helpstring("property Count")]
 HRESULT Count([out, retval] long *pVal);
 [id(2), helpstring("method Add")]
 HRESULT Add([in] short ChannelNum, [out, retval] IIntDChannel
 **pVal);
 [propget, id(3), helpstring("property Item")]
 HRESULT Item([in] short ChannelNum, [out,retval] IIntDChannel*
 *pVal);
 [id(4), helpstring("method Remove")]
 HRESULT Remove([in] short ChannelNum);
 [id(5), helpstring("method RemoveAll")]
 HRESULT RemoveAll();
 [id(0xfffffff), propget, restricted, hidden, helpstring("property _newEnum")]
 HRESULT _newEnum([out, retval] IUnknown* *pVal);
};
```

**Property: \_newEnum** (Interface IEnumVARIANT): This property is implemented by using rules of the COM environment, and provides “for each” functionality.

**Property: Count:** Contains a list length.

**Property Item:** Returns an IIntDChannel object for the particular channel number.

**Method: Add**

Inserts a new channel into the channel list and returns a newly created IIntDChannel object

**Method: Remove**

Removes a particular channel from the channel list.

**Method: RemoveAll**

Removes all channels from the channel list.

## IIntDChannel Interface

The IIntDChannel interface provides access to an IIntDChannel object.

**Interface Definition**

```
[
 object,
 uuid(F6B8C61B-C92D-41AE-AD42-377D7F29D776),
 dual,
 helpstring("IIntDChannel Interface"),
 pointer_default(unique)
]
interface IIntDChannel : IDispatch
{
 [propget, id(1), helpstring("property RecorderId")]
 HRESULT RecorderId([out, retval] BSTR *pVal);
 [propget, id(2), helpstring("property ChannelNum")]
 HRESULT ChannelNum([out, retval] short *pVal);
 [propget, id(3), helpstring("property Cuts")]
 HRESULT Cuts([out, retval] IIntDCutList/*IUnknown*/ *pVal);
};
```

**Property: RecorderId:** Contains a recorder name.

**Property: ChannelNum:** Contains a channel number.

**Property: Cuts:** Returns an IIntDCutSeries object.

## IIntDCutSeries Interface

The IIntdCutSeries interface provides basic navigation through the list of IIntDPureCut objects. It extends the COM interface IEnumVARIANT that provides ability to use “for each” statements in Visual Basic and JScript.

### Interface Definition

```
[
 object,
 uuid(08E6B6BF-9EC6-4F3C-B4D0-374A968FADA7),
 dual,
 helpstring("IIntdCutSeries Interface"),
 pointer_default(unique)
]
interface IIntDCutSeries : IDispatch {
 [id(0xffffffffc), propget, restricted, hidden]
 HRESULT _NewEnum([out, retval] IUnknown** ppUnk);

};
```

**Property: \_newEnum** (Interface IEnumVARIANT): This property is implemented by using rules of the COM environment, and provides “for each” functionality.

## IIntDCutList Interface

The IIntdCutList interface extends the IIntDCutSeries interface.

### Interface Definition

```
[
 object,
 uuid(0C6126A6-523E-4D95-A7CA-DE326F7656A2),
 dual,
 helpstring("IIntDCutList Interface"),
 pointer_default(unique)
]
interface IIntDCutList : IIntDCutSeries
{
 [propget, id(1), helpstring("property Count")]
 HRESULT Count([out, retval] long *pVal);
 [id(2), helpstring("method Add")]
 HRESULT Add([in] IIntDPureCut *Cut, [out, retval] IIntDPureCut
 **pVal);
 [id(3), helpstring("method RemoveAll")]
 HRESULT RemoveAll();
};
```

**Property: Count:** Contains list length.

### Method: Add

Inserts new a pure cut into a cut list and returns a newly created IIntDPureCut object. In addition, if the data source supports auditing, an archived copy of the pure cut will also be created.

### Method: RemoveAll

Removes all cuts from cut list.

## IIntDCutMessageList Interface

The IIntDCutMessageList interface provides basic navigation through the list of cut messages.

### Interface Definition

```
[
 object,
```

```
 uuid(64E86328-13FB-4E22-8D5D-88A2024B09E3),
 dual,
 helpstring("IIntDCutMessageList Interface"),
 pointer_default(unique)
]
 interface IIntDCutMessageList : IDispatch
 {
 [id(0xffffffffc), propget, restricted, hidden, helpstring("property _newEnum")]
 HRESULT _newEnum([out, retval] IUnknown* *pVal);
 [id(1), helpstring("method AddMessage")]
 HRESULT AddMessage([in] BSTR Message);
 };
```

**Property \_newEnum** (Interface IEnumVARIANT): This property is implemented by using rules of the COM environment, and provides “for each” functionality.

#### **Method: AddMessage**

This method adds a new message into a message list.

## **IIntDCutMessage Interface**

The IIntDMessage interface provides access to an IIntDMessage object.

#### **Interface Definition**

```
[
 object,
 uuid(D2D37216-2B87-49A7-A4E9-10D4227ED2BD),
 dual,
 helpstring("IIntDCutMessage Interface"),
 pointer_default(unique)
]
interface IIntDCutMessage : IDispatch
{
 [propget, id(1), helpstring("property Message")]
 HRESULT Message([out, retval] BSTR *pVal);
};
```

**Property: Message:** Contains a message string.

## IntervalList Interface

The IIntervalList interface provides basic navigation through a list of interval indexes. Indexes are zero based.

### Interface Definition

```
[
 object,
 uuid(ECFE6996-143A-4a96-8A91-514887053CF2),
 dual,
 helpstring("IIntervalList Interface"),
 pointer_default(unique)
]
interface IIntervalList : IDispatch
{
 [propget, id(1), helpstring("property Count")]
 HRESULT Count([out, retval] long *pVal);
 [id(2), helpstring("method Item")]
 HRESULT Item([in] long Index, [out, retval] long *IntervalIndex);
};
```

**Property: Count:** Contains list length.

### Method: Item

This method returns an interval index from particular list item.

## IIntDPureCut Interface

The IIntDPureCut interface provides access to an IIntDPureCut object.

### Interface Definition

```
[
 object,
 uuid(4495B4E5-3AED-4EFB-A447-4AD92D5EF2AF),
 dual,
 helpstring("IIntDPureCut Interface"),
 pointer_default(unique)
]
interface IIntDPureCut : IDispatch
{
 [propget, id(1), helpstring("property StartTime")]
 HRESULT StartTime([out, retval] DATE *pVal);
 [propput, id(1), helpstring("property StartTime")]
 HRESULT StartTime([in] DATE newVal);

 [propget, id(2), helpstring("property StopTime")]
 HRESULT StopTime([out, retval] DATE *pVal);

 [propget, id(4), helpstring("property SPI")]
 HRESULT SPI([out, retval] long *pVal);
 [propput, id(4), helpstring("property SPI")]
 HRESULT SPI([in] LONG newVal);

 [propget, id(5), helpstring("property RecorderId")]
 HRESULT RecorderId([out, retval] BSTR *pVal);
 [propput, id(5), helpstring("property RecorderId")]
 HRESULT RecorderId([in] BSTR newVal);

 [propget, id(6), helpstring("property ChannelNum")]
 HRESULT ChannelNum([out, retval] short *pVal);
 [propput, id(6), helpstring("property ChannelNum")]
 HRESULT ChannelNum([in] short newVal);

 [propget, id(7), helpstring("property ValidationMessages")]
```

```

 HRESULT ValidationMessages([out, retval] IIntDCutMessageList
**pList);

[propget, id(8), helpstring("property TrailMessages")]
 HRESULT TrailMessages([out, retval] IIntDTrailMessageList **pList);

[propget, id(9), helpstring("property Descriptor")]
 HRESULT Descriptor([out, retval] BSTR *pVal);
[propput, id(9), helpstring("property Descriptor")]
 HRESULT Descriptor([in] BSTR newVal);

[propget, id(10), helpstring("property Edited")]
 HRESULT Edited([out, retval] VARIANT_BOOL *pVal);
[propput, id(10), helpstring("property Edited")]
 HRESULT Edited([in] VARIANT_BOOL newVal);

[propget, id(11), helpstring("property ExternallyValid")]
 HRESULT ExternallyValid([out, retval] VARIANT_BOOL *pVal);

[propget, id(12), helpstring("property InternallyValid")]
 HRESULT InternallyValid([out, retval] VARIANT_BOOL *pVal);

[propget, id(13), helpstring("property ReadyToMerge")]
 HRESULT ReadyToMerge([out, retval] VARIANT_BOOL *pVal);
[propput, id(13), helpstring("property ReadyToMerge")]
 HRESULT ReadyToMerge([in] VARIANT_BOOL newVal);

[propget, id(14), helpstring("property Origin")]
 HRESULT Origin([out, retval] BSTR *pVal);
[propput, id(14), helpstring("property Origin")]
 HRESULT Origin([in] BSTR newVal);

[propget, id(15), helpstring("property UOMCode")]
 HRESULT UOMCode([out, retval] short *pVal);
[propput, id(15), helpstring("property UOMCode")]
 HRESULT UOMCode([in] SHORT newVal);

[propget, id(16), helpstring("property UID")]
 HRESULT UID([out, retval] BSTR *pVal);

[id(50), helpstring("method Load")]
 HRESULT Load(
 [in, defaultvalue(intdLoadValMessages | intdLoadTrails)]IntDLoadFlags
flags,
 [out, retval] IIntDCut **pVal);

[id(51), helpstring("method Remove")]
 HRESULT Remove();

[id(52), helpstring("method Replace")]
 HRESULT Replace([in] IIntDPureCut *Cut, [in] IntDSaveFlags flags);

[id(53), helpstring("method Restore")]
 HRESULT Restore([out, retval] IIntDPureCut **pCut);

[propget, id(54), helpstring("property OriginalDataSource")]
 HRESULT OriginalDataSource([out, retval] BSTR* pVal);

[propput, id(54), helpstring("property OriginalDataSource")]
 HRESULT OriginalDataSource([in] BSTR newVal);
[propget, id(55), helpstring("property Modified")]
 HRESULT Modified([out, retval] VARIANT_BOOL* pVal);
[propget, id(56), helpstring("property TZSTDName")]
 HRESULT TZSTDName([out, retval] BSTR* pVal);
[propput, id(56), helpstring("property TZSTDName")]
 HRESULT TZSTDName([in] BSTR newVal);
[id(57), helpstring("method SaveToXML")]

```

```
HRESULT SaveToXML([in]VARIANT Stream, [out, retval] BSTR* Dest);
};
```

**Property: RecorderId:** Contains the Recorder id.

**Property: ChannelNum:** Contains the Channel number.

**Property: StartTime:** Contains the StartTime of cut.

**Property: StopTime:** Contains the StopTime of cut.

**Property: SPI:** Contains the SPI of cut.

**Property: ValidationMessages:** Returns pointer on an IIntDTrailMessageList interface (look at IIntDTrailMessageList)

**Property: TrailMessages:** Returns pointer on IIntDTrailMessageList interface (look at IIntDTrailMessageList)

**Property: Descriptor:** Contains the Descriptor of cut.

**Property: Edited:** Contains the Edited flag of cut.

**Property: ExternallyValid:** Contains the External validation flag of cut.

**Property: InternallyValid:** Contains the Internal validation flag of cut.

**Property: ReadyToMerge:** Contains the Ready to merge flag of cut.

**Property: Origin:** Contains the Origin flag of cut.

**Property: UOMCode:** Contains the UOM code.

**Property: UID:** Contains the unique identifier of cut for current data source.

**Property: OriginalDataSource:** Contains/allows setting the original datasource of cut(implemented only in WSETProvider).

**Property: Modified:** Contains the Modified flag.

**Property: TZSTDName:** Contains/allows setting time zone standard name.

#### **Method: Load**

Loads interval data and returns a newly created IIntDCut object;

#### **Method: Restore**

Restores a pure cut from archive if the audit is supported by a provider and turned on. This method returns the restored cut from an archived IIntDPureCut object.

#### **Method: Remove**

Removes the cut from datasource.

**Method: Replace**

Replaces a pure cut in the cut list and returns a newly created IIntDPureCut object. If the data source supports the EditTrail functionality, an old cut will be updated according to the EditTrail rules.

**Method: SaveToXML**

Serializes a cut in XML format (see XMLProvider)

Parameter Stream contains the output object. The following VARIANT types are supported

- VT\_EMPTY, XML will be returned as BSTR string (BSTR\* Dest)
- VT\_UNKNOWN, which can contain a value in the form of an IStream, ISequentialStream. XML will be saved by using these interfaces (UTF-8 encoding).

**Enum IntDLoadFlags:** contains type of loading for Load method.

```
typedef [uuid(78785933-336A-4410-A3B8-DA16EC8D57B9), version(1.0)]
enum {
 intdLoadNothing = 0,
 intdLoadValMessages = 1,
 intdLoadTrails = 2,
 intdLoadAsIs = 0,
 intdCheckOverCount = 4,
 intdFillMissing = 8,
 intdCheckUnderCount = 16
} IntDLoadFlags;
```

Value	Description
intdLoadNothing	Do not load any messages
intdLoadValMessages	Load validation messages
intdLoadTrails	Load trail messages
intdLoadAsIs	Load cut as is
intdCheckOverCount	if an interval count more than an expected count returns error
intdFillMissing	Fill missing for DST period if an interval count less than an expected count
intdCheckUnderCount	if an interval count less than an expected count returns error

**Enum IntDSaveFlags:** contains type of saving for Replace and Add methods.

```
typedef [uuid(FA6B7B51-FAE4-4be9-A882-DDFD1264B911), version(1.0)]
enum {
 intdSaveSkipValMessages = 0,
 intdSaveAppendValMessages = 1,
 intdSaveOverValMessages = 2,
 intdSaveSkipTrails = 0,
 intdSaveAppendTrails = 4,
 intdSaveOverTrails = 8
} IntDSaveFlags;
```

Value	Description
intdSaveSkipValMessages	Skip validation messages
intdSaveAppendValMessages	Append validation messages to existing(up to 10)
intdSaveOverValMessages	Overwrite validation messages

IIntDCut Interface

The IIntDCut interface extends the IIntDPureCut interface. Interfaces and methods that appear in bold face are published, others are internal interfaces.

Interface Definition

```
[
 odl,
 uuid(8F340037-5D4E-47c3-99E1-A934D5B90ADE),
 version(1.0),
 dual,
 oleautomation
]
interface IIntDCut : IIntDPureCut {
 [id(0x60020000)]
 HRESULT Scale(
 [in] IntDPeriod period,
 [in] IntDAggregationType aggtype,
 [out, retval] IIntDCut** pRetVal);

 [id(0x60020001)]
 HRESULT ExportXml(
 [in] IntDXmlFormat XmlFormat,
 [in] IntDXmlDataExport DataExport,
 [in] BSTR DateFormat,
 [out, retval] BSTR* pRetVal);

 [id(0x60020002)]
 HRESULT BlockOp(
 [in] IntDBlockOperation blockType,
 [in] IIntDCut* pIntDCut,
 [out, retval] IIntDCut** pRetVal);

 [id(0x60020003)]
 HRESULT BlockOpNA(
 [in] IntDBlockOperation blockType,
 [in] IIntDCut* pIntDCut,
 [out, retval] IIntDCut** pRetVal);

 [id(0x60020004)]
 HRESULT Clone ([out, retval] IIntDCut** pRetVal);

 [id(0x60020005), helpstring("method CopyCut")]
 HRESULT CopyCut([in] BSTR RecorderId, [in] long ChannelId, [in] DATE
 StartTime, [out,
 retval] IIntDCut **pVal);

 [id(0x60020006), helpstring("method IntDCount")]
```

```

HRESULT IntDCount([in] IntDTypeCount type, [out, retval] long *ret);

[id(0x60020007), helpstring("method Value")]
HRESULT Value([in] long index, [out, retval] IIntDCutValue **pVal);

[id(0x60020008), helpstring("method SetValueStatus")]
HRESULT SetValueStatus([in, defaultvalue(intdCOMP_NONE)] IntDCompOperation
op,
 [in] BSTR status,
 [in] BSTR newStatus,
 [in] VARIANT start,
 [in] VARIANT stop,
 [in, optional] VARIANT newValue,
 [out, retval] IIntDCut **pRet);
[id(0x60020009), helpstring("method SetString")]
HRESULT SetString([in] BSTR Status, [out, retval] IIntDCut **pRet);
[id(0x60020010), helpstring("method RemoveSpikes")]
HRESULT RemoveSpikes([in] long SequenceLength,
 [in] double maxValue,
 [in, defaultvalue("K")] BSTR NewStatus);
[id(0x60020011), helpstring("method RemoveDips")]
HRESULT RemoveDips([in] long SequenceLength,
 [in] double minValue,
 [in, defaultvalue("K")] BSTR Status);
[id(0x60020012)]
HRESULT Smooth(
 [in] IntDSmoothType smoothType,
 [in, optional] VARIANT value,
 [out, retval] IIntDCut** pRetVal);

[id(0x60020013), helpstring("method Interpolate")]
HRESULT Interpolate([in] VARIANT Start,
 [in] VARIANT Stop,
 [in, defaultvalue("8")] BSTR QualityCode,
 [in, defaultvalue("J")] BSTR NewStatus);
[id(0x60020014), helpstring("method SpikeTest")]
HRESULT SpikeTest([in] long lPeaks,
 [in] double dPercent,
 [in, defaultvalue("9")] BSTR StatusCode,
 [out, retval] IIntervalList **pRet);
[id(0x60020015), helpstring("method DipTest")]
HRESULT DipTest([in] long lDips,
 [in] double dPercent,
 [in, defaultvalue("9")] BSTR StatusCode,
 [out, retval] IIntervalList **pRet);
[id(0x60020016), helpstring("method Join")]
HRESULT Join([in] IIntDCut *pCut,
 [in] IntDMergeType mrType,
 [in] BOOL bMatchUOM,
 [in] BOOL bValidate,
 [out, retval] IIntDCut **pRet);
[id(0x60020017), helpstring("method Prorate")]
HRESULT Prorate([in, defaultvalue(1.0e100)] double highBound,
 [in, defaultvalue(-1.0e100)] double lowBound,
 [in, defaultvalue("")] BSTR Status,
 [in, optional] VARIANT meterEnergy,
 [in, defaultvalue("")] BSTR newStatus,
 [out, retval] IIntDCut **pVal);
[id(0x60020018)]
HRESULT ScalarOp(
 [in] IntDScaleOperation scaleType,
 [in] VARIANT value,
 [in, defaultvalue("L")] BSTR Status,
 [out, retval] IIntDCut** pRetVal);

[id(0x60020019)]
HRESULT ScalarOpRange(
 [in] IntDScaleOperation scaleType,

```

```
[in] VARIANT value,
[in] DATE StartTime,
[in] DATE StopTime,
[in, defaultvalue("L")] BSTR Status,
[out, retval] IIntDCut** pRetVal);

[id(0x60020020)]
HRESULT Split(
 [in] IntDSplitType splitType,
 [in] DATE SplitTime,
 [out, retval] IIntDCut** pRetVal);

[id(0x60020021)]
HRESULT ReplaceRange(
 [in] double dValue,
 [in] DATE StartTime,
 [in] DATE StopTime,
 [in, defaultvalue("L")] BSTR Status,
 [out, retval] IIntDCut** pRetVal);

[id(0x60020022)]
HRESULT DeleteIntervals(
 [in] DATE StartTime,
 [in] DATE StopTime,
 [out, retval] IIntDCut** pRetVal);

[id(0x60020023)]
HRESULT InsertIntervals(
 [in] double dValue,
 [in] DATE StartTime,
 [in] DATE StopTime,
 [in, defaultvalue("J")] BSTR Status,
 [out, retval] IIntDCut** pRetVal);

[id(0x60020024)]
HRESULT CalcStopTime();

[id(0x60020025)]
HRESULT KnownMeterReading(
 [in] DATE MeterTime,
 [in] double dMeterReading,
 [in, defaultvalue(0)] long lDials,
 [in, defaultvalue(0)] long lDecimals);

[id(0x60020027), helpstring("method ShiftStartTime")]
HRESULT ShiftStartTime(
 [in] IntDShiftType shiftType,
 [in] DATE StartTime);

[id(0x60020028)]
HRESULT InsertIntervalsFromCut(
 [in] DATE StartTime,
 [in] DATE StopTime,
 [in] IIntDCut* pIntDCut,
 [in, defaultvalue("01/01/1970 00:00:00")] DATE StartTime2,
 [in, defaultvalue("**")] BSTR Status,
 [out, retval] IIntDCut** pRetVal);

[id(0x60020029)]
HRESULT ReplaceRangeFromCut(
 [in] DATE StartTime,
 [in] DATE StopTime,
 [in] IIntDCut* pIntDCut,
 [in, defaultvalue("01/01/1970 00:00:00")] DATE StartTime2,
 [in, defaultvalue("**")] BSTR Status,
 [out, retval] IIntDCut** pRetVal);

[id(0x60020030), helpstring("method SetStatuses")]
```

```

HRESULT SetStatuses([in] VARIANT Start, [in] VARIANT Stop, [in] BSTR
Status);

[id(0x60020031), helpstring("method SliderValuesCSV")]
HRESULT SliderValuesCSV([in] LONG Start, [in] LONG Amount, [out,retval]
BSTR* Values);

[id(0x60020032), helpstring("method SliderStatusesCSV")]
HRESULT SliderStatusesCSV([in] LONG Start,
[in] LONG Amount, [out,retval] BSTR* Statuses);
[id(0x60020033), helpstring("method SliderDatesCSV")]
HRESULT SliderDatesCSV([in] LONG Start, [in] LONG Amount,
[in, optional] VARIANT fDateformat, [in, optional] VARIANT fTimeformat,
[in, defaultvalue(" ")] BSTR Separator, [out,retval] BSTR* Dates);

[propget, id(0x60020113), helpstring("property ValuesCSV")]
HRESULT ValuesCSV([out, retval] BSTR *pVal);
[propget, id(0x60020114), helpstring("property StatusesCSV")]
HRESULT StatusesCSV([out, retval] BSTR *pVal);
[propget, id(0x60020115), helpstring("property DatesCSV")]
HRESULT DatesCSV(BSTR DataFormat, [out, retval] BSTR *pVal);

[propget, id(1610744087), helpstring("property DSTParticipant")]
HRESULT DSTParticipant([out, retval] BSTR* pVal);
[propput, id(1610744087), helpstring("property DSTParticipant")]
HRESULT DSTParticipant([in] BSTR newVal);
[propget, id(1610744090), helpstring("property TimeStamp")]
HRESULT TimeStamp([out, retval] VARIANT* pVal);
[propput, id(1610744090), helpstring("property TimeStamp")]
HRESULT TimeStamp([in] VARIANT newVal);
[propget, id(1610744091), helpstring("property UOMName")]
HRESULT UOMName([out, retval] BSTR* pVal);
[propget, id(1610744092), helpstring("property UOMAggregate")]
HRESULT UOMAggregate([out, retval] BSTR* pVal);
[propget, id(1610744093), helpstring("property MaxTime")]
HRESULT MaxTime([out, retval] VARIANT* pVal);
[propget, id(1610744094), helpstring("property Maximum")]
HRESULT Maximum([out, retval] DOUBLE* pVal);
[propget, id(1610744095), helpstring("property PulseMultiplier")]
HRESULT PulseMultiplier([out, retval] DOUBLE* pVal);
[propput, id(1610744095), helpstring("property PulseMultiplier")]
HRESULT PulseMultiplier([in] DOUBLE newVal);
[propget, id(1610744096), helpstring("property FlagE")]
HRESULT FlagE([out, retval] BSTR* pVal);
[propget, id(1610744097), helpstring("property FlagI")]
HRESULT FlagI([out, retval] BSTR* pVal);
[propget, id(1610744098), helpstring("property FlagN")]
HRESULT FlagN([out, retval] BSTR* pVal);
[propget, id(1610744099), helpstring("property FlagO")]
HRESULT FlagO([out, retval] BSTR* pVal);
[propget, id(1610744100), helpstring("property Weight")]
HRESULT Weight([out, retval] DOUBLE* pVal);
[propput, id(1610744100), helpstring("property Weight")]
HRESULT Weight([in] DOUBLE newVal);
[propget, id(1610744101), helpstring("property Population")]
HRESULT Population([out, retval] DOUBLE* pVal);
[propput, id(1610744101), helpstring("property Population")]
HRESULT Population([in] DOUBLE newVal);
[propget, id(1610744102), helpstring("property AbsMaxTime")]
HRESULT AbsMaxTime([out, retval] VARIANT* pVal);
[propget, id(1610744103), helpstring("property AbsMaximum")]
HRESULT AbsMaximum([out, retval] DOUBLE* pVal);
[propget, id(1610744104), helpstring("property KWMax")]
HRESULT KWMax([out, retval] DOUBLE* pVal);
[propget, id(1610744105), helpstring("property Minimum")]
HRESULT Minimum([out, retval] DOUBLE* pVal);
[propget, id(1610744106), helpstring("property MinTime")]
HRESULT MinTime([out, retval] VARIANT* pVal);

```

```

 [propget, id(1610744107), helpstring("property MinimumNZ")]
 HRESULT MinimumNZ([out, retval] DOUBLE* pVal);
 [propget, id(1610744108), helpstring("property CountNZ")]
 HRESULT CountNZ([out, retval] LONG* pVal);
 [propget, id(1610744109), helpstring("property CountNon9Val")]
 HRESULT CountNon9Val([out, retval] LONG* pVal);
 [propget, id(1610744110), helpstring("property Total")]
 HRESULT Total([out, retval] DOUBLE* pVal);
 [propget, id(1610744111), helpstring("property Energy")]
 HRESULT Energy([out, retval] DOUBLE* pVal);
 [propget, id(1610744112), helpstring("property Average")]
 HRESULT Average([out, retval] DOUBLE* pVal);
 [propget, id(1610744113), helpstring("property AverageNZ")]
 HRESULT AverageNZ([out, retval] DOUBLE* pVal);
 [propget, id(1610744114), helpstring("property MaximumN")]
 HRESULT MaximumN([in, defaultvalue(1)] LONG Index, [out, retval] DOUBLE*
pVal);
 [propget, id(1610744115), helpstring("property MaxTimeN")]
 HRESULT MaxTimeN([in, defaultvalue(1)] LONG Index, [out, retval] VARIANT*
pVal);
 [propget, id(1610744126), helpstring("property IPH")]
 HRESULT IPH([out, retval] LONG* pVal);
 [propget, id(1610744129), helpstring("property DeleteFlag")]
 HRESULT DeleteFlag([out, retval] VARIANT_BOOL* pVal);
 [propget, id(1610744129), helpstring("property DeleteFlag")]
 HRESULT DeleteFlag([in] VARIANT_BOOL newVal);
 [propget, id(1610744130), helpstring("property EditedRulsLang")]
 HRESULT EditedRulsLang([out, retval] VARIANT_BOOL* pVal);
 [propget, id(1610744131), helpstring("property LoadFactor")]
 HRESULT LoadFactor([out, retval] DOUBLE* pVal);
 [propget, id(1610744132), helpstring("property MeterStartReading")]
 HRESULT MeterStartReading([out, retval] DOUBLE* pVal);
 [propget, id(1610744132), helpstring("property MeterStartReading")]
 HRESULT MeterStartReading([in] DOUBLE newVal);
 [propget, id(1610744133), helpstring("property MeterStopReading")]
 HRESULT MeterStopReading([out, retval] DOUBLE* pVal);
 [propget, id(1610744133), helpstring("property MeterStopReading")]
 HRESULT MeterStopReading([in] DOUBLE newVal);
 [propget, id(1610744134), helpstring("property MeterOffset")]
 HRESULT MeterOffset([out, retval] DOUBLE* pVal);
 [propget, id(1610744134), helpstring("property MeterOffset")]
 HRESULT MeterOffset([in] DOUBLE newVal);
 [propget, id(1610744135), helpstring("property MeterMultiplier")]
 HRESULT MeterMultiplier([out, retval] DOUBLE* pVal);
 [propget, id(1610744135), helpstring("property MeterMultiplier")]
 HRESULT MeterMultiplier([in] DOUBLE newVal);
 [propget, id(1610744136), helpstring("property Timezone")]
 HRESULT Timezone([out, retval] SHORT* pVal);
 [propget, id(1610744136), helpstring("property Timezone")]
 HRESULT Timezone([in] SHORT newVal);
 [propget, id(1610744137), helpstring("property UOMTotalize")]
 HRESULT UOMTotalize([out, retval] BSTR* pVal);
 [propget, id(1610744138), helpstring("property UOMRateQuantity")]
 HRESULT UOMRateQuantity([out, retval] BSTR* pVal);
 [propget, id(1610744139), helpstring("property UOMRelatedCode")]
 HRESULT UOMRelatedCode([out, retval] SHORT* pVal);
 [propget, id(1610744140), helpstring("property PulseOffset")]
 HRESULT PulseOffset([out, retval] DOUBLE* pVal);
 [propget, id(1610744140), helpstring("property PulseOffset")]
 HRESULT PulseOffset([in] DOUBLE newVal);
 };

```

Published Properties are self evident and are properties of the Cut Object. Some may be set, some are read only. See **Appendix B: Oracle Utilities Enhanced Input/Output Interval Data Format** in the *Data Manager User's Guide* for details on individual properties.

**Published Method: Scale**

This method is encapsulated within the IIntDCut interface. It creates and returns the address of a new IIntDCut object, where each interval has the SPI of the specified period. It aggregates the intervals of the current cut, based on the aggregation type (either add or average).

The first argument sent to the method is the period. The second argument sent to the method is the aggrtype. A newly created IIntDCut object conforming to the request is returned to the calling procedure.

**Published Method: Split**

This method splits a cut in two at the time you specify.

```
HRESULT Split(
 [in] IntDSplitType splitType,
 [in] DATE SplitTime,
 [out, retval] IIntDCut** pRetVal);
```

This method accepts 2 arguments and returns the first or second cut based on the splitType. A newly created IIntDCut object conforming to the request is returned to the calling procedure.

- **splitType.** This parameter specifies whether the first or second cut should be returned.
- **SplitTime:** A Date Value representing the time that the original cut will be split. The time entered must be between the start and stop times of the original cut. This time becomes the start time of the second cut created by the split. A time of 1 second before this is calculated as the stop time of the first cut.
- **Enum IntDSplitType:** contains the split operations that can be performed on cuts.

```
typedef [uuid(e576dc30-a022-41ce-86aa-50a9e37bb01d), version(1.0)]
enum {
 intdSPLIT_FIRST = 0,
 intdSPLIT_SECOND = 1
} IntDSplitType;
```

Value	Description
intdSPLIT_FIRST	Return first split cut.
intdSPLIT_SECOND	Return second split cut.

**Published Method: CalcStopTime**

This method calculates the stop time of the cut based on the start time of the cut, the number of intervals, the DST\_Participant flag, and the seconds-per-interval.

```
HRESULT CalcStopTime();
```

The re-computed stop-time is stored in the cut. This method is used to fix an incorrectly transcribed stop-time, or to force the cut's expected intervals to match the recorded intervals.

## IIntDCutValue Interface

The IIntdCutValue interface provides access to values and status codes in the Cut. It is created via the Value method of the IIntDCut object.

**Interface Definition**

```
[
 odl,
 uuid(9ED54F84-A89D-4FCD-A854-44251E925F09),
 version(1.0),
 dual,
 oleautomation
]
interface IIntdCutValue : IDispatch {
[id(0x60020000), propget]
 HRESULT Value([out, retval] double* pRetVal);
[id(0x60020000), propput]
 HRESULT Value([in] double pRetVal);
[id(0x60020001), propget]
 HRESULT Status([out, retval] BYTE* pRetVal);
[id(0x60020001), propput]
 HRESULT Status([in] BYTE pRetVal);
[id(0x60020002), propget]
 HRESULT DateTime([out, retval] DATE* pRetVal);
[id(0x60020002), propput]
 HRESULT DateTime([in] DATE* pRetVal);
};
```

**Property: Value:** Allows users to set and get of the values or intervals.

**Property: Status:** Allows users to set and get the status codes.

## IIntDProvider Interface

The IIntDProvider interface allows a connection to a data source provider.

```
[
 odl,
 uuid(02F59F2D-B197-4b77-BB55-C17742C5379F),
 version(1.0),
 dual,
 oleautomation
]
interface IIntDProvider : IDispatch {
 [id(0x60020000)]
 HRESULT Connect([in] BSTR ConnectString);
};
```

### Method: Connect

The IIntDDataSource object uses this method to connect a provider to a particular data source.

# Interval Data Source Providers

## RDBProvider

The RDBProvider provides ability to use a relational database as an interval datasource. The RDBProvider is the COM component that implements the interface of interval datasource provider.

**ProgID:** LSIntD.RDBProvider

**Provider-specific connection string:** RDBProvider accepts following XML structure as the connection string:

```
<PROVIDER PROGID="LSIntD.RDBProvider">
 <DATASOURCE>
 . . . LODESTAR's DATASOURCE structure
 </DATASOURCE>
 [<CUTHEADERTABLE>string</CUTHEADERTABLE>]
</PROVIDER>
```

This XML structure contains the following XML elements:

- **DATASOURCE:** (Required) Standard DATASOURCE structure. Default value is N/A
- **CUTHEADERTABLE:**(Optional) Contains the name of a table in the data source which contains headers of cuts. Default value is LSCHANNELCUTHEADER

## BTEProvider

This provider will allow for access to interval data in a Pervasive database. (filename.bte)

**ProgID:** LSIntD.BTEProvider

**Provider-specific connection string:** The BTEProvider accepts following XML structure as the Provider string:

```
<PROVIDER PROGID="LSIntD.BTEProvider"
 PATH="String"
</PROVIDER>
```

This XML structure contains the following XML elements:

- **PATH:** (Required) Fully qualified path the BTE file set. Default value is N/A.

## DATASOURCE Definition

The DTD for a valid DATASOURCE XML structure is shown below.

```
<!DOCTYPE DATASOURCE
[
<!ELEMENT DATASOURCE (NAME, CONNECTSTRING?, USERID?, PASSWORD?,
 QUALIFIER?)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT CONNECTSTRING (#PCDATA)>
<!ELEMENT USERID (#PCDATA)>
<!ELEMENT PASSWORD (#PCDATA)>
<!ELEMENT QUALIFIER (#PCDATA)>
]>
```

The first time that a DataSource object is initialized with a unique data source name, a static structure will be cached to store information related to the data source. The first data source that is cached will become the default data source. Any subsequent initialization of a DataSource object with a previously used data source name will create an object representing the previously cached structure.

**Element Descriptions:** The only required argument for DataSource initialization is the data source name (except for the default constructor). The ConnectString argument must be provided prior to actually creating a Connection object. The first UserID, Password, or Qualifier argument provided for each unique data source will become the default for that data source.

## Examples

This example DATASOURCE XML string is an example of how the XML should look if connecting to a BTE file using many defaulted settings.

```
<INTDATASOURCE>
<PROVIDER
 PROGID='LSIntD.BTEProvider'
 PATH='\"d:\btefiles\mybtefile.bte\"'>
</PROVIDER>
<SESSION/>
</INTDATASOURCE>
```

This example DATASOURCE XML string is an example of how the XML should look if connecting to a RDB Interval data store.

```
<INTDATASOURCE>
 <PROVIDER PROGID="LSIntD.RDBProvider">
 <DATASOURCE>
 <NAME>INTDTEST</NAME>
 <CONNECTSTRING>DSN=MYODBC;UID=MYUSERID;PWD=MYPASSWORD;</CONNECTSTRING>
 <QUALIFIER>MYQUALIFIER</QUALIFIER>
 </DATASOURCE>
 <CUTHEADERTABLE>LSCHANNELCUTHEADER</CUTHEADERTABLE>
 </PROVIDER>
 <SESSION>
 <INTD_NEW_JOIN INTDMERGESELECT="RECENT" CHECKVALID="Y"/>
 <CHECK_UOM_MATCHING CHECK_COMPATIBILITY="Y"/>
 </SESSION>
</INTDATASOURCE>
```

## Sample Application using LSINTD

This section contains a sample application that uses the LSINTD component.

```
Private Sub cmdExtractAll_Click()
' This subroutine is not very optimized.
' You may not want to load node details unless user clicks a node.
 Dim FileName As String 'Filename of the BTE file
 Dim objectIntd As LSINTDLib.IntDDDataSource 'the datasource object
 Dim objRecorder As LSINTDLib.IIntDRecorder '
 Dim objChannel As LSINTDLib.IIntDChannel
 Dim objCuts As LSINTDLib.IIntDCutList
 Dim objPureCut As LSINTDLib.IIntDPureCut
 Dim ObjCut As LSINTDLib.IIntDCut
 Dim objIntervalValue As LSINTDLib.IIntDCutValue
 Dim iIntervalCount As Integer
 Dim itemp As Integer
 Dim iIndex As Long
 Dim dblIntervalValue As Double
 Dim strIntervalStatusValue As String
 Dim strOutput As String
 Dim iNumProcessed As Long
 Dim strReference As String
 Dim strChanReference As String

 On Error GoTo ErrorHandler
 Me.tvwOutput.Visible = True
 Me.lstOutput.Visible = False
 Me.txtOutput.Visible = False
 'prepare
 FileName = txtFileName.Text
 If FileName = "" Then Exit Sub
 cmdExit.Enabled = False
 glbCancelLoad = False
 'strOutput = ""
 Me.tvwOutput.Nodes.Clear
 Me.txtTotalLoaded.Text = "0"
 Me.pbLoadProgress.Visible = True
 Me.lblProgress.Visible = True

 'instantiate the datasource object and create the session
 'Set ObjectIntd = CreateObject("LSIntD.IntdDataSource")
 Set objectIntd = New LSINTDLib.IntDDDataSource
 objectIntd.Connect "<INTDATASOURCE><PROVIDER PROGID='LSIntD.BTEProvider'
 PATH='" + _ FileName + "'><USERID/></PROVIDER><SESSION/></INTDATASOURCE>"

 'Set the progress bar
 Me.pbLoadProgress.Max = objectIntd.Recorders.Count
 Me.txtTotalRecorders = CStr(objectIntd.Recorders.Count)
 iNumProcessed = 1

 'loop through the collection of recorders in the datasource object
 For Each objRecorder In objectIntd.Recorders
 DoEvents 'Allows for cancel button to interrupt
 If glbCancelLoad = True Then
 Exit For
 End If

 Me.pbLoadProgress.Value = iNumProcessed
 Me.txtTotalLoaded.Text = CStr(iNumProcessed)

 'Get properties and display to user for recorder level
 'Add the recorder ID as a node
 Me.tvwOutput.Nodes.Add , , objRecorder.RecorderId, objRecorder.RecorderId
 & " [" & objRecorder.Channels.Count & " channels]"

 'Loop through the collection of channels in the recorder object
 For Each objChannel In objRecorder.Channels
```

```

'Add a channel node
strChanReference = objRecorder.RecorderId & CStr(objChannel.ChannelNum)
Me.tvwOutput.Nodes.Add objRecorder.RecorderId, tvwChild,
strChanReference, objChannel.ChannelNum

'Cuts is a cutlist object that contains cuts from the channel
'Get cuts from the channel and set to Cuts object
Set objCuts = objChannel.Cuts

'We have a list of all LodeStar cuts in the channel object
'Loop through the cuts and display info about them (properties)
For Each objPureCut In objCuts

 'strOutput = strOutput + vbTab & vbTab & "Cut : " &
objPureCut.StartTime & " - " & objPureCut.StopTime & vbCrLf
 'Create the cut Status object
 On Error Resume Next
 Set ObjCut = objPureCut.Load() 'use debug to view properites
 If Err.Number <> 0 Then
 GoTo ErrorLabel
 End If
 On Error GoTo ErrorHandler

 'Add a cut node for a pure cut
 strReference = objRecorder.RecorderId & CStr(objChannel.ChannelNum) &
CStr(objPureCut.StartTime)
 Me.tvwOutput.Nodes.Add strChanReference, tvwChild, strReference,
objPureCut.StartTime

 'Add Cut information:
 strOutput = "StartTime: " & CStr(ObjCut.StartTime)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "StopTime: " & CStr(ObjCut.StopTime)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "StopTime: " & CStr(ObjCut.StopTime)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "Meter Start: " & CStr(ObjCut.MeterStartReading)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "Meter Stop: " & CStr(ObjCut.MeterStopReading)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "Meter Mult: " & CStr(ObjCut.MeterMultiplier)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "Pulse Mult: " & CStr(ObjCut.PulseMultiplier)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "Pulse Offset: " & CStr(ObjCut.PulseOffset)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "Interval Total: " & CStr(ObjCut.Total)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "Population: " & CStr(ObjCut.Population)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "Weight: " & CStr(ObjCut.Weight)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "SPI: " & CStr(ObjCut.SPI)
 Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

 strOutput = "Interval Count: " & CStr(ObjCut.IntervalCount)

```

```
Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

strOutput = "UOM: " & CStr(ObjCut.UOMCode)
Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

strOutput = "TsWoGMT: " & CStr(ObjCut.Timezone)
Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

strOutput = "Descriptor: " & CStr(ObjCut.Descriptor)
Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

strOutput = "Validation Messages " & "[" &
ObjCut.ValidationMessages.Count & "]"
Dim strVmsgParent As String
strVmsgParent = objRecorder.RecorderId & CStr(objChannel.ChannelNum)
& CStr(objPureCut.StartTime) & "V"
Me.tvwOutput.Nodes.Add strReference, tvwChild, strVmsgParent,
strOutput
'Add vmsg here as child key =
If ObjCut.ValidationMessages.Count >= 1 Then

 iIndex = 1
 Do While iIndex <= ObjCut.ValidationMessages.Count
 'DoEvents
 strOutput = "V Message #" & CStr(iIndex) & " = " &
ObjCut.ValidationMessages(iIndex - 1).Text
 Me.tvwOutput.Nodes.Add strVmsgParent, tvwChild, , strOutput
 iIndex = iIndex + 1
 Loop
End If

strOutput = "TZSN: " & CStr(ObjCut.TZSTDName)
Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

strOutput = "Edited: " & CStr(ObjCut.Edited)
Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

strOutput = "Edited: " & CStr(ObjCut.Edited)
Me.tvwOutput.Nodes.Add strReference, tvwChild, , strOutput

'Prepare for values and status codes for the cut.
iIntervalCount = ObjCut.IntervalCount
itemp = 1
'This loop gets the actual intervals and status codes of the pure cut.
Do While itemp <= iIntervalCount
 Set objIntervalValue = ObjCut.Value(itemp)
 dblIntervalValue = objIntervalValue.Value 'not actually doing
anything with the values and statuses
 strIntervalStatusValue = objIntervalValue.Status
 itemp = itemp + 1
Loop
ErrorLabel:
 Set ObjCut = Nothing
Next
Set objCuts = Nothing
Next
iNumProcessed = iNumProcessed + 1
Next

'txtOutput.Text = strOutput
'Release the objects
Me.MousePointer = vbNormal
Set objectIntd = Nothing
Set objRecorder = Nothing
Set objChannel = Nothing
Set objCuts = Nothing
Set ObjCut = Nothing
```

```

 Set objPureCut = Nothing
 Set objIntervalValue = Nothing
 Me.pbLoadProgress.Visible = False
 Me.lblProgress.Visible = False
 cmdExit.Enabled = True
 Exit Sub
 ErrorHandler:

 Me.pbLoadProgress.Visible = False
 Me.lblProgress.Visible = False
 cmdExit.Enabled = True
 Me.MousePointer = vbNormal
 MsgBox "Error No.: " & Err.Number & vbCrLf & "Description: " & Err.Description

 End Sub

Private Sub cmdExtractCut_Click()
 Dim FileName As String 'Filename of the BTE file
 Dim objectIntd As LSINTDLib.IntDDDataSource 'the datasource object
 Dim objPureCut As LSINTDLib.IIntDPureCut
 Dim ObjCut As LSINTDLib.IIntDCut
 Dim objIntervalValue As LSINTDLib.IIntDCutValue
 Dim iIntervalCount As Integer
 Dim iIndex As Integer
 Dim itemp As Integer
 Dim dblIntervalValue As Double
 Dim strIntervalStatusValue As String
 Dim strOutput As String
 Dim datStartDate As Date
 Dim DateSupplied As Boolean
 Dim ReturnCode As Long

 Me.tvwOutput.Visible = False
 Me.lstOutput.Visible = False
 Me.txtOutput.Visible = True
 On Error GoTo ErrorHandler
 'Set up...
 Me.txtTotalRecorders = ""
 Me.txtTotalLoaded = ""
 glbCancelLoad = False
 'Me.txtOutput.Text = ""
 'Me.MousePointer = vbHourglass
 strOutput = ""
 FileName = txtFileName.Text
 If FileName = "" Then Exit Sub

 'instantiate the datasource object and create the session
 'Set ObjectIntd = CreateObject("LSIntD.IntdDataSource")
 Set objectIntd = New LSINTDLib.IntDDDataSource
 objectIntd.Connect "<INTDATASOURCE><PROVIDER PROGID='LSIntD.BTEProvider'
 PATH='" + FileName + "'><USERID/></PROVIDER><SESSION/></INTDATASOURCE>"
 DateSupplied = True
 If Not IsDate(Me.txtStartTime) Then
 DateSupplied = False
 MsgBox ("Invalid date.")
 Exit Sub
 Else
 datStartDate = CDate(Me.txtStartTime)
 'Get the Pure Cut
 Set objPureCut = objectIntd.GetCutByKey(UCase(Me.txtRecorderID),
 CLng(Me.txtChannel), datStartDate)
 End If
 'Example of Loading a cut over a date range
 'Set ObjCut = objectIntd.LoadDates(Recorder, Channel, StartDate, StopDate)

 Set ObjCut = objPureCut.Load()

 'Format the string of cut attributes for display

```

```

ReturnCode = FormatOutput(ObjCut, strOutput)
Me.txtOutput.Text = strOutput
'Prepare for values and status codes for the cut.

'This loop gets the actual intervals and status codes of the pure cut.
'These are only for viewing inb the debugger.
'Not outputted to the results control.
'Illustrates how to get individual intervals and statuses
'Uncomment the following Lines if required
'iIntervalCount = ObjCut.IntervalCount
'itemp = 1
'Do While itemp <= iIntervalCount
' Set objIntervalValue = ObjCut.Value(itemp)
' dblIntervalValue = objIntervalValue.Value 'not actually doing anything
 with the values and statuses
' strIntervalStatusValue = objIntervalValue.Status
' itemp = itemp + 1
'Loop

'Release the objects
Me.MousePointer = vbNormal
Set objectIntd = Nothing
Set objPureCut = Nothing
Set objIntervalValue = Nothing
Exit Sub
ErrorHandler:
Me.MousePointer = vbNormal
MsgBox "Error No.: " & Err.Number & vbCrLf & "Description: " & Err.Description
End Sub

Function FormatOutput(ObjCut As LSINTDLib.IIntDCut, strOutput As String) As Long

Dim iIndex As Long

strOutput = strOutput & "RecID: " & vbTab & vbTab & ObjCut.RecorderId & vbCrLf
strOutput = strOutput & "Channel: " & vbTab & vbTab & ObjCut.ChannelNum &
vbCrLf
strOutput = strOutput & "Start: " & vbTab & vbTab & ObjCut.StartTime & vbCrLf
strOutput = strOutput & "Stop: " & vbTab & vbTab & ObjCut.StopTime & vbCrLf
strOutput = strOutput & "Met Start: " & vbTab & ObjCut.MeterStartReading &
vbCrLf
strOutput = strOutput & "Met Stop: " & vbTab & ObjCut.MeterStopReading &
vbCrLf
strOutput = strOutput & "Met Mult: " & vbTab & vbTab & ObjCut.MeterMultiplier
& vbCrLf
strOutput = strOutput & "Int Total: " & vbTab & vbTab & ObjCut.Total & vbCrLf
strOutput = strOutput & "SPI: " & vbTab & vbTab & ObjCut.SPI & vbCrLf
strOutput = strOutput & "UOM: " & vbTab & vbTab & ObjCut.UOMCode & vbCrLf
strOutput = strOutput & "Desc: " & vbTab & vbTab & ObjCut.Descriptor & vbCrLf
strOutput = strOutput & "Num ValMsgs: " & vbTab &
ObjCut.ValidationMessages.Count & vbCrLf

If ObjCut.ValidationMessages.Count > 0 Then
iIndex = 1
Do While iIndex <= ObjCut.ValidationMessages.Count
strOutput = strOutput & "ValMsg" & iIndex & ": " & _
vbTab & ObjCut.ValidationMessages.Item(iIndex - 1).Text &
vbCrLf
iIndex = iIndex + 1
Loop
End If

strOutput = strOutput & "TZSN: " & vbTab & vbTab & ObjCut.TZSTDName & vbCrLf
strOutput = strOutput & "Edited: " & vbTab & vbTab & ObjCut.Edited & vbCrLf
strOutput = strOutput & "Int Val: " & vbTab & vbTab & ObjCut.InternallyValid
& vbCrLf

```

```
strOutput = strOutput & "Ext Val: " & vbTab & vbTab & ObjCut.ExternallyValid
& vbCrLf
strOutput = strOutput & "Merge: " & vbTab & vbTab & ObjCut.ReadyToMerge &
vbCrLf
strOutput = strOutput & "Archv: " & vbTab & vbTab & ObjCut.DeleteFlag & vbCrLf
strOutput = strOutput & "Origin: " & vbTab & vbTab & ObjCut.Origin & vbCrLf
strOutput = strOutput & "DST Flag: " & vbTab & ObjCut.DSTParticipant & vbCrLf
strOutput = strOutput & "Edited By RS: " & vbTab & ObjCut.EditedRulsLang &
vbCrLf

strOutput = strOutput & vbCrLf
strOutput = strOutput & "STATISTICS:" & vbCrLf
strOutput = strOutput & "Energy: " & vbTab & vbTab & ObjCut.Energy & vbCrLf

strOutput = strOutput & "Avg Int: " & vbTab & vbTab & ObjCut.Average & vbCrLf
strOutput = strOutput & "Avg NZ: " & vbTab & vbTab & ObjCut.AverageNZ & vbCrLf
strOutput = strOutput & "Peak: " & vbTab & vbTab & ObjCut.AbsMaxTime & vbCrLf
FormatOutput = 0
Exit Function
End Function
```

## INTDVB Compatibility

This section describes corresponding interfaces and methods in the LSINTD component to the functions in the INTDVB component.

To begin using the LSINTD.DLL in VB, the programmer must add the Reference to the VB Project. This will allow for the use of the Interface in the MS IDE.

### INTDVB - LSINTD Mapping

INTDVB Function Name	LSINTD Interface, Method
IntDataOpenSession	<pre>Dim objectIntd As LSINTDLib.IntDDDataSource 'the datasource object  Set objectIntd = New LSINTDLib.IntDDDataSource objectIntd.Connect "&lt;INTDATASOURCE&gt;&lt;PROVIDER PROGID='LSIntD.BTEProvider' PATH='" + FileName + "'&gt;&lt;USERID/&gt;&lt;/ PROVIDER&gt;&lt;SESSION/&gt;&lt;/ INTDATASOURCE&gt;"</pre>
IntDataGetError	<p>This function is not necessary. Errors may be obtained from the Err object.</p> <p>Use an Error Handler.</p>
IntDataCloseSession	<p>LSINTD method not needed. Set the Object to Nothing.</p> <p>Set objectIntd = Nothing</p>
IntDataGetCutHeaders	<pre>Dim objRecorder As LSINTDLib.IIntDRecorder ' Dim objChannel As LSINTDLib.IIntDChannel Dim objCuts As LSINTDLib.IIntDCutList Dim ObjCut As LSINTDLib.IIntDCut  Dim objIntervalValue As LSINTDLib.IIntDCutValue  For Each objRecorder In objectIntd.Recorders For Each objChannel In objRecorder.Channels Set objCuts = objChannel.Cuts  'We have a list of all LodeStar cuts in the channel object  'Loop through the cuts and display info about them (properties)</pre>

INTDVB Function Name	LSINTD Interface, Method
IntDataGetCutHeaderValue	<p>For Each objPureCut In objCuts</p> <p>'Load the Cut's info and Properties</p> <p>Set ObjCut = objPureCut.Load()</p> <p>strReference = objRecorder.RecorderId &amp; _  CStr(objChannel.ChannelNum) &amp; _  CStr(objPureCut.StartTime)</p>
IntDataReleaseCutHeaderArray	Not Needed. Set the Objects to Nothing.
IntDataLoadCut	<p>Dim objectIntd As LSINTDLib.IntDDDataSource</p> <p>'the datasource object</p> <p>Dim objPureCut As LSINTDLib.IIntDPureCut</p> <p>Dim ObjCut As LSINTDLib.IIntDCut</p> <p>Dim objIntervalValue As  LSINTDLib.IIntDCutValue</p> <p>Set objectIntd = New LSINTDLib.IntDDDataSource</p> <p>objectIntd.Connect</p> <p>"&lt;INTDATASOURCE&gt;&lt;PROVIDER  PROGID='LSIntD.BTEProvider' PATH='" +  FileName + "'&gt;&lt;USERID/&gt;&lt;/  PROVIDER&gt;&lt;SESSION/&gt;&lt;/  INTDATASOURCE&gt;"</p> <p>Set objPureCut =  objectIntd.GetCutByKey(UCase(Me.txtRecorderID)  , _  CLng(Me.txtChannel), datStartDate)</p> <p>Set ObjCut = objPureCut.Load()</p> <p>"This loop gets the actual intervals and status codes  of the pure cut.</p> <p>'Illustrates how to get individual intervals and  statuses</p> <p>iIntervalCount = ObjCut.IntervalCount</p> <p>itemp = 1</p> <p>Do While itemp &lt;= iIntervalCount</p> <p>Set objIntervalValue = ObjCut.Value(itemp)</p> <p>dblIntervalValue = objIntervalValue.Value</p> <p>strIntervalStatusValue = objIntervalValue.Status</p> <p>itemp = itemp + 1</p> <p>Loop</p>
IntDataLoadCutEx	<p>Set ObjCut = objectIntd.LoadDates(Recorder,  Channel, StartDate, _ StopDate)</p>

INTDVB Function Name	LSINTD Interface, Method
IntDataSaveCut	objPureCut = objectIntd.Cuts.Add(objPureCut, 0) objPureCut = ObjCut.Replace(objPureCut, 0)
IntDataReleaseCut	Set the Object to Nothing
IntDataGetCutIntervals	Not needed. See IntDataLoadCut
IntDataGetCutValues	Not needed. See IntDataLoadCut
IntDataSubset	Perform two calls to the Split Method. ObjCut1 = ObjCut.Split(0, ObjCut) ObjCut2 = ObjCut.Split(1, ObjCut)
IntDataDelete	This was an undocumented interface in INTDVB. Use the Remove Method of LSINTD. ObjCut.Remove

# Part Four

---

## Appendices

Part Four includes appendices that describe import and export file formats used by the Oracle Utilities Energy Information Platform, and contains the following chapters:

- **Appendix A: Oracle Utilities Data Repository Database Schema**
- **Appendix B: Oracle Utilities Data Repository Schema Import File Format**
- **Appendix C: Oracle Utilities Enhanced Input/Output Interval Data Format**
- **Appendix D: Oracle Utilities Standard XML Interval Data Format**
- **Appendix E: Oracle Utilities Comma Separated Values (CSV) Interval Data Format**

---

# Appendix A

---

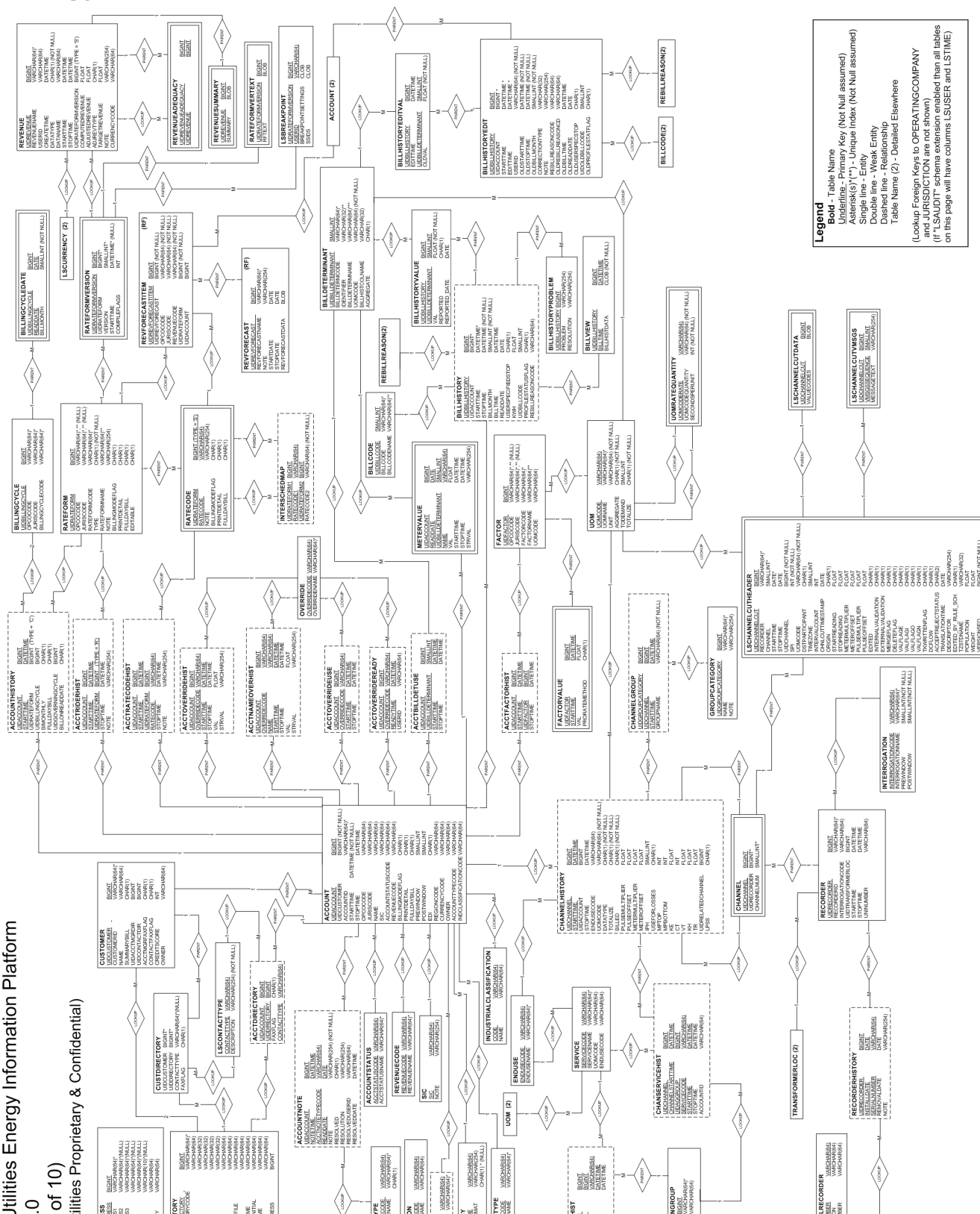
## Oracle Utilities Data Repository Database Schema

This appendix includes a diagram of the Oracle Utilities Data Repository database schema (v1.6.1.0.0) that provides details regarding the table and columns in the schema, as well as the relationships between these tables in the Oracle Utilities Data Repository. This information is very useful when writing Rules Language statements or constructing database queries. This includes:

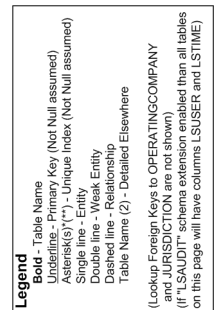
- **Energy Information Platform Schema p. 1**
- **Energy Information Platform Schema p. 2**
- **Energy Information Platform Schema - Security**
- **Energy Information Platform Schema - Reporting**
- **Energy Information Platform Schema - Messaging**
- **Energy Information Platform Schema - Work Queues**
- **Energy Information Platform Schema - Interval Data Manager**
- **Energy Information Platform Schema - Adapter**
- **Energy Information Platform Schema - Service Points and Market Participants**
- **Energy Information Platform Schema - Web Services**

## Energy Information Platform Schema p. 1

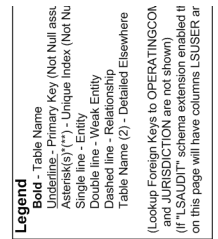
Utilities Energy Information Platform  
0 of 10  
ilities Proprietary & Confidential)







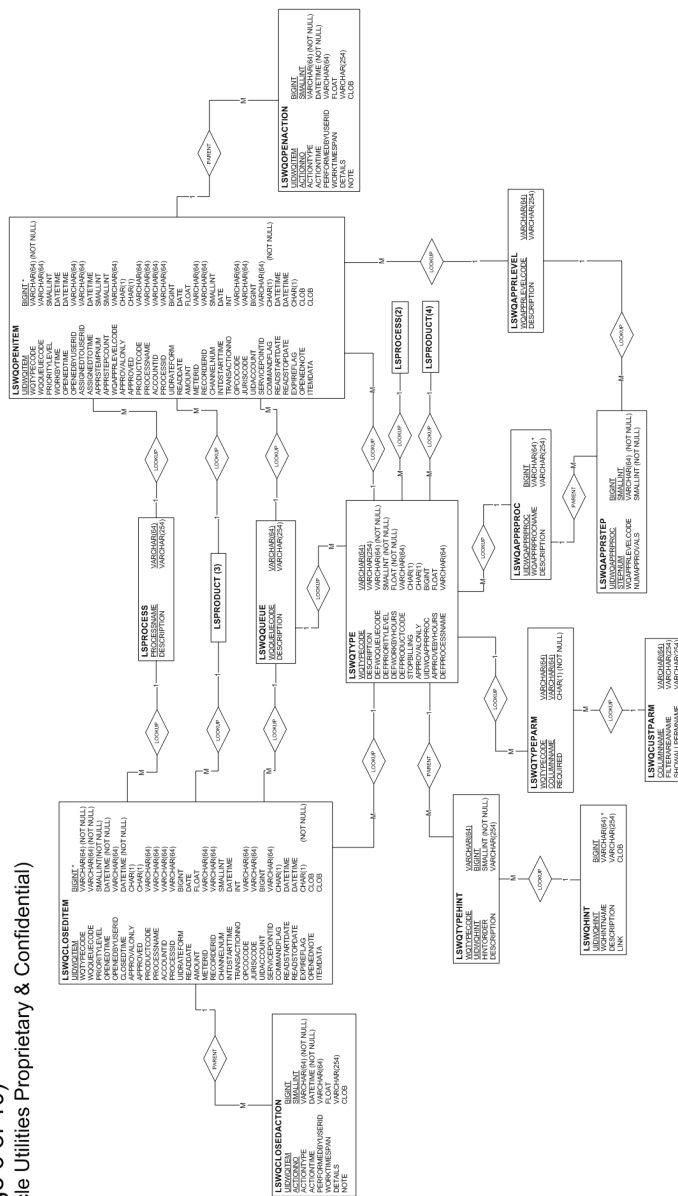
## Oracle Utilities Data Repository Database Schema A-5





## Energy Information Platform Schema - Work Queues

Oracle Utilities Energy Information Platform  
v1.6.0.0 – Work Queues  
Page 6 of 10)  
Oracle Utilities Proprietary & Confidential)

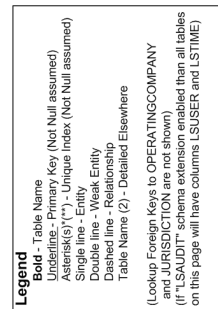


### Legend

**Legend**

- Bold** - Table Name
- Underline - Primary Key (Not Null assure)
- Asterisk(s)(\*) - Unique Index (Not Null)
- Single line - Entity
- Double line - Weak Entity
- Dashed line - Relationship
- Table Name (2) - Detailed Elsewhere

(Lookup Foreign Keys to OPERATINGCOMMI and JURISDICTION are not shown)  
(If "LSAUDIT" schema extension enabled then on this page will have columns LSUSER and



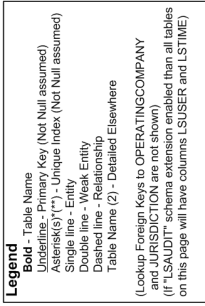
## Energy Information Platform Schema - Adapter



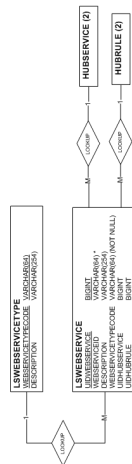
**Legend**

<b>Bold</b> - Table Name
<u>Underline</u> - Primary Key (Not Null assure)
Asterisks(*) - Unique Index (Not Null)
Single line - Entity
Double line - Weak Entity
Dashed line - Relationship
Table Name (2) - Detailed Elsewhere

(LookUp Foreign Keys to OPERATINGCO, and JURISDICTION are not shown) (If LAUSDIT schema extension enabled if on this page will have columns LAUSER ar



## Energy Information Platform Schema - Web Services



**Legend**

<b>Bold</b>	Table Name
<u>Underscore</u>	Primary Key (Not Null assure Asterisk(s)***)
Single line	Entity
Dashed line	Weak Entity
Dashed line	Relationship
Table Name (2)	Detailed Elsewhere

(Lookup Foreign Keys to OPERATINGCOM and JURISDICTION are not shown)  
(If "LRUSDIT" schema extension enabled then on this page will have columns LUSER an



# Appendix B

---

## Oracle Utilities Data Repository Schema Import File Format

This appendix describes the format of the input files required by the Oracle Utilities import utilities, based on the base Oracle Utilities Data Repository schema (LSDB 5.31.002), including:

- **Record Formats**
  - **A Note Regarding Dependent Records**
- **Oracle Utilities Import Format (\*.imp)**
- **Oracle Utilities XML Import Format (\*.xml)**

### **Important Note**

Any client-specific customizations to this base schema, whether implemented by the client or Oracle, may alter one or more of the definitions provided below.

## Record Formats

Import files can be in one of two formats:

- **Oracle Utilities Import Format (\*.imp)**
- **Oracle Utilities XML Import Format (\*.xml)**

### A Note Regarding Dependent Records

Dependent records can only be added to the database after all of their parent records have been added. If a dependent record and one or more of its parent records are loaded from the same file, the parent records must appear before the dependent record in the file.

## Oracle Utilities Import Format (\*.imp)

The Oracle Utilities Import Format is a comma-delimited format in which each column value for a record must appear on a single line, in correct order, separated by commas. The following rules provide specific details concerning the input file format:

- Each record must begin on a new line (though text wrapping may occur).
- Comments may be placed on a line by themselves by beginning the line with an asterisk (\*) character. Blank lines are also allowed.
- The first value of each record must be the identifying mnemonic for the record type. The table layouts (see **How to view or print database table layouts:** on page B-3) contain the mnemonic for each table.
- Each value may optionally be enclosed by double-quotation marks. If a value is enclosed in double quotation marks, then all characters between the quotation marks will be considered part of the value, including any leading or trailing blanks.
- If a value contains a comma (,) character, or any leading or trailing blank characters, then it **must** be enclosed in double-quotation marks.
- Null values are represented by no value. Trailing null values must be preceded by commas.

### Example:

\*The following lines contain the Customer, Account Status, Jurisdiction, Operating Company, Revenue Code, and Account information.

```
CUST,"800001","Paul Pham",,"HOME","OFFICE1","E","E"
ACCTSTAT,"A","Excellent"
JURIS,"SF","Smithfield"
OPCO,"GECO","Generic Electric Company","123 Mars Planet
Ave","Universe","GK","12345-4321","991-123-4567"
REVCODE,"33","Commercial (Light)"
ACCT,"800001","800001","01/01/1993 00:00:00",,"GECO","SF","Paul
Sauvageau",,"A","33","2","3","N",5,5
```

Each table in the Oracle Utilities Data Repository has its own record format, based on the specific columns in the table. The PLIMPORT command line program can be used to view or output a layout of each table that defines its record format.

The -t switch on the PLIMPORT program allows users to view the layout of a table (or tables), or to create a text-only output file that contains the table layout. See **Importing Customer Data** on page 8-4 of the *Oracle Utilities Energy Information Platform Configuration Guide* for more information about using the PLIMPORT program.

**Note:** Using the -t switch overrides the other parameters. PLIMPORT can either import data or display tables in the database, but not both in the same command line. In other words, if you use the -t switch in the command line, PLIMPORT will NOT import data.

### How to view or print database table layouts:

1. Open a Command Prompt (DOS) Window. Select **Start->Run**, and type 'cmd' in the field that appears.
2. Change to the directory in which your Oracle Utilities executables are stored, typically \LODESTAR\BIN.
3. To view the database layout, type the following command at the prompt:

```
PLIMPORT -d dsn -t TABLENAME
```

where:

- The **-d** switch is identical to the switch described in **Importing Customer Data** on page 8-4 of the *Oracle Utilities Energy Information Platform Configuration Guide*.
  - The **-t** switch specifies the table you wish to view. To view the entire database, enter **-t** without the tablename parameter.
4. To save the table layouts to a text file, type the following command at the prompt:

```
PLIMPORT -d dsn -t TABLENAME > outputfile.txt
```

where:

- **outputfile.txt** is name of the output file containing the table layouts.

PLIMPORT writes the text-only output file containing the layout of the specified table(s) to the current directory, which can be viewed and printed using Notepad or another text editor.

For example, to save the layout of the Account Table in the “lodestar” database to a file named acctpict.txt, type the following

```
PLIMPORT -d lodestar -t ACCOUNT > acctpict.txt
```

The output might look like the following:

Column Name	Data Type	Data Length	Req'd?	Unique	Identity	Foreign Key
MNEMONIC	“ACCT”	N/A	X			
CUSTOMERID	character	64 (max)	X			1 - CUSTOMER
ACCOUNTID	character	64 (max)	X	1	X	
STARTTIME	datetime	N/A	X			
STOPTIME	datetime	N/A				
OPCOCODE	character	64 (max)				2 - OPERATINGCOMPANY
JURISCODE	character	64 (max)				3 - JURISDICTION
NAME	character	64(max)				
SIC	character	64 (max)				4 - SIC
ACCOUNTSTATUSCODE	character	64 (max)				5 - ACCOUNTSTATUS
REVENUECODE	character	64 (max)				6 - REVENUECODE
BILLINGMODEFLAG	character (0,1,2,3)	1				
PRINTDETAIL	character (0,1,2,3)	1				
FULLDAYBILL	character (0,Y, N)	1				
PREWINDOW	small integer	N/A				
POSTWINDOW	small integer	N/A				
EDI	character	1				
REGIONCODE	character	64 (max)				7 - REGION

## Rules

There are specific rules regarding the Oracle Utilities Import format, including:

- A single record will have at least one unique column (or compound column combinations), representing the primary key, or identity, of the record.
- The use of a tilde (“~”) in an identity column causes the column to be ignored. The identity of the record must still be discernible from the included identity columns.
- A record may have zero or more foreign key columns (or compound column combinations). The existence of a foreign key indicates a dependent record, meaning that all of its parent records must exist in the database before it is added. If a dependent record and one or more of its parent records are imported in the same file, the parent records must appear before the dependent record in the file. For example, before records can be added to the Account table in the above example, appropriate records must exist in the database in the Customer, Operating Company, Jurisdiction, SIC, Account Status, Revenue Code, and Region tables. If these records were imported using the same import file, the Customer, Operating Company, Jurisdiction, SIC, Account Status, Revenue Code, and Region table records would have to appear before the Account table records in the import file.
- If a record is added and one of the columns in the record is Null (i.e. there is no value for the column provided in the input file), the column is ignored. This allows default values in the database to remain unaffected.
- If updating a record, the new record must be followed **on the same line** by a comma and the primary key of the existing record (comma separated if more than one column).

### Example:

Update the jurisdiction record from Smithfield (SF) to California (CA).

```
JURIS,"CA","California",, "SF"
```

## Date Formats

*Billmonth* data types are formatted as “MM/YYYY”.

*Date* data types are formatted as either:

- “MM/DD/YYYY”
- “YYYY/MM/DD”

*Datetime (Timestamp)* data types are formatted as either:

- “MM/DD/YYYY HH:MM:SS”
- “YYYY/MM/DD HH:MM:SS”

**Note:** If import data contains milliseconds, the milliseconds must be set to “000” before importing.

## Oracle Utilities XML Import Format (\*.xml)

The Oracle Utilities XML Import Format is a XML format in which each record and column value for a record are defined in a specific XML structure. This is the same XML format used by the Oracle Utilities Database interface (LSImportDB). A Data Type Definition (DTD), xml examples, and data element descriptions for this format are provided below.

### DTD - Oracle Utilities XML Import Format

```
<!DOCTYPE LODESTAR_IMPORT
[
<!ELEMENT LODESTAR_IMPORT (ROWS*)>
<!ATTLIST LODESTAR_IMPORT
 ACTION#PCDATA,
 BATCHSIZE#PCDATA,
 ONERROR#PCDATA>
<!ELEMENT ROWS(ROW*)>
<!ATTLIST ROWS
 TBL #PCDATA>
<!ELEMENT ROW (DBCOLUMNNAME)>
<!ATTLIST ROW
 TBL CDATA #IMPLIED,
 ACTION(ADD|UPDATE|DELETE|ADDUPDATE)#IMPLIED>
<!ELEMENT DBCOLUMNNAME(#PCDATA)>
<!ATTLIST DBCOLUMNNAME
 V CDATA "column value",
 NV CDATA "new column value">
]>
```

### XML Examples - Oracle Utilities XML Import Format

#### Add Example

```
<LODESTAR_IMPORT ACTION="ADD" BATCHSIZE="50" ONERROR="CONTINUE">
 <ROWS>
 <ROW TBL="ACCOUNT">
 <CUSTOMERID V="Cust1"/>
 <ACCOUTNID V="Acct1"/>
 <STARTTIME V="01/01/2000 00:00:00"/>
 </ROW>
 </ROWS>
</LODESTAR_IMPORT>
```

#### Update Example

```
<LODESTAR_IMPORT ACTION="UPDATE" BATCHSIZE="50" ONERROR="STOP">
 <ROWS TBL="ACCOUNT">
 <ROW>
 <CUSTOMERID V="Cust1"/>
 <ACCOUTNID V="Acct1" NV="Acct100"/>
 <STARTTIME V="01/01/2001 00:00:00"/>
 </ROW>
 </ROWS>
</LODESTAR_IMPORT>
```

## Element Descriptions - Oracle Utilities XML Import Format

**LODESTAR\_IMPORT:** Root element containing one or more sets of records, each defined in a ROWS element.

Attributes:

**ACTION:** Defines the action performed. Possible values are:

- **ADD:** Adds one or more records to the database. It is an error if the record(s) is already present.
- **UPDATE:** Updates one or more existing records in the database. It is an error if the record does not exist.
- **DELETE:** Deletes one or more records in the database. It is an error if the record to be deleted does not exist.
- **ADDUPDATE:** Adds or Updates one or more records in the database. This is the Default behavior. If a record is present, the function updates it. If the record does not exist, the function adds the record.

**BATCHSIZE:** The number of records to be sent to the database in a single call. Larger numbers generally improve performance. The default is 50. The maximum is 500.

**ONERROR:** Defines how errors are handled. Possible values include:

- **CONTINUE:** Continue importing records after error.
- **STOP:** Stop importing records when an error occurs.

Elements:

**ROWS:** Element containing one or more records for import, each defined in a ROW element.

Attributes:

**TBL:** Table name for all the records defined in the ROW elements included in the ROW element.

Elements:

**ROW:** Element containing one record for import.

Attributes:

**TBL:** Table name for the record. If present in a ROW element, this overrides the TBL attribute on the ROWS element.

Elements:

**DBCOLUMNNAME:** Actual database name of the column. For example, when importing records in the Account ID column of the Account table, this would be "ACCOUNTID".

Attributes:

**V:** Column value. If not present, NULL is assumed. If the value is an XML, it will be specified as a child of the DBCOLUMNNAME element.

**NV:** New column value for the record. Must be specified if updating a key column.

**Notes:**

- The "NV" (New value) attribute is only required for identity columns that need to be updated.
- Also, for "DELETE" operations, providing just the identity columns is sufficient.
- Any missing "V" (Value) attribute will be treated as a NULL value for the column.



# Appendix C

---

## Oracle Utilities Enhanced Input/Output Interval Data Format

This document provides a detailed description of the Oracle Utilities Enhanced Input/Output Interval Data (LSE) format, used by Oracle Utilities applications. This includes:

- **Enhanced Format**
- **Enhanced Format Details**
- **Sample Files**
- **Units of Measure**

# Enhanced Format

## General Field Descriptions

In the enhanced format, fields are comma-delimited. A data line may contain as many full data-status-time groups as desired. No physical record in the file may exceed a character count (record length) of 32750. Leading or trailing white space is allowed on any field, but will be ignored upon processing. *The descriptor field is an exception to this rule.* It may contain any number of spaces, anywhere within the field.

Numeric fields will be expressed as decimal numbers, and may contain a decimal point and/or a leading minus sign (-), but must not contain commas or the value “+” to indicate a positive number. Non-negative numbers are assumed on all numeric fields if no leading minus sign is present.

The Customer Identifier, Channel Number, and the Start Time compose the full key of the record which, in combination with each other, must uniquely identify the record to be stored in the database.

No field except the descriptor may contain a comma.

All character values in the Enhanced Format must be uppercase.

Any cut entered with an invalid Start or Stop Time will be rejected.

An input cut with header records, but no data records, will be treated like any other cut; the number of intervals allocated will be determined by the time range, and all intervals will be treated as missing.

The contents of the Customer Identifier field, Origin field, and all Y/N indicators will be translated to uppercase before they are stored in the database.

**Note:** Database sort order will follow ASCII collating sequence, starting with the Customer Identifier and followed in turn by the Channel, then the Start Time.

A cut will be rejected if the number of interval values supplied is greater than the number of interval values expected, based upon the Start Time, Stop Time, SPI, and DST\_Participant fields. If too few interval values are supplied, the cut will be filled out with “missings” (value of 0 and status of ‘9’).

Any Stop Time falling exactly on an interval boundary will have one second subtracted from it.

## Field Relationships and Requirements

Relationships between fields exist in the enhanced format. These are described along with detailed field requirements in this section. Defaults are inserted when the processing program encounters an omitted field (indicated by double commas) or white space between commas.

### Sort Code

#### Relationships:

This field has no relationship with any other field.

#### Requirements:

- Sort codes must contain eight digits, including leading 0s. Sort codes are restricted to integers between 00000001 and 99999999. However, 00000005 through 09999999 are not used, and may be ignored (see below).
- Leading white space before the sort codes is not allowed.
- The first three header records are required.
- The fourth header record is optional.
- Header records will have sort code 00000001 through 00000004.
- Column 1 of each header record must contain a 0.
- Records with Sort codes 00000005 through 05000000 are reserved.
- Sort codes 00000030 through 00000039 are reserved for user-defined attributes created through Oracle Utilities Rules Language rate schedules.
- Data records must start with sort code 10000000, and progress sequentially (i.e. 10000001, 10000002...) through 99999999, or as high as needed to accommodate the data, with the maximum of 99999999.
- Column 1 of the data records must be greater than 0.

### Recorder/Customer Identifier (key element)

#### Relationships:

None.

#### Requirements:

- Only letters, numbers, underscores, and hyphens will be accepted in this field; all other characters and embedded blanks will cause the record to be rejected.  
**Note:** Lowercase letters will be translated to uppercase before they are stored in the database.
- Must be at least one character, and no more than 64 characters in length.

#### Default:

None; field is required.

**Channel (key element)****Relationships:**

None.

**Requirements:**

- Field values must be non-negative integer only.
- Maximum value is 32767.

**Default:**

None; field is required.

**Start Time (key element)****Relationships:**

- Must be a valid datetime value less than the value of the Stop Time.

**Requirements:**

- Format must be YYYYMMDDHHMMSS.
- Value must be after 19700101000000 when using Data Manager, Oracle Utilities Billing Component, and other non-Oracle Utilities Load Analysis applications. The value must be after 19670101000000 when using Oracle Utilities Load Analysis.
- No spaces, colons, or any ASCII value allowed between time components.

**Default:**

None; field is required.

**Stop Time****Relationships:**

- Must be a valid datetime value greater than the value of the Start Time.
- Value must agree with the number of interval values past the start time \* SPI.
- One second is subtracted if the value is equal to an even interval start time.

**Requirements:**

See Start Time.

**Default:**

None; field is required

**DST Participant Flag****Relationships:**

None.

**Requirements:**

Valid values are Y, N, or A, indicating whether or not this record will be processed using DST adjustments.

---

Flag Value	Definition
Y	(DST Participant) The intervals recorded participate in DST adjustments and the intervals during DST reflect the time change.

---

Flag Value	Definition
N	(Does not participate in DST) The intervals recorded do not make any DST time changes.
A	(DST participant) The intervals have been adjusted to 24-hour days. (At the April time change, there should be a value at the time change (a place holder) of 0 with status code “9”. At the October time change, there should be a combined value.) These cuts will be converted to a “Y” or “N”, and are not stored in the database as “A”.

**Default:**

As specified in the INTDDEFAULTDST configuration parameter setting, the INTDCONFIG.XML file, or the CSDST value in Oracle Utilities Load Analysis.

**Invalid Record Flag****Relationships:**

None.

**Requirements:**

Valid values are Y or N, indicating whether or not the input programs should validate the incoming record.

Flag Value	Definitions
Y	The incoming record contains unvalidated data. In Oracle Utilities Load Analysis, a value of Y can be used to run this record through the validation routines.
N	The incoming record contains valid data (mark as valid).

**Default:**

Y.

**Meter Start Reading****Relationships:**

Must be supplied if any of the other Meter fields (Meter Stop Reading, Meter Multiplier, or Meter Offset) are supplied.

**Requirements:**

If supplied, value must be a non-negative numeric value.

**Default:**

0.

**Meter Stop Reading****Relationships:**

Must be supplied if any of the other Meter Fields (Meter Start Reading, Meter Multiplier, or Meter Offset) are supplied.

**Requirements:**

If supplied, value must be a non-negative numeric value.

**Default:**

0 stored in the database if not supplied.

**Meter Multiplier**

**Relationships:**

Must be supplied if any of the other Meter fields (Meter Start Reading, Meter Stop Reading, or Meter Offset) are supplied.

**Requirements:**

If supplied, value must be positive numeric. However, if no value is specified, 0 will be stored. Zero (0) is not a valid value for the meter multiplier, but its presence in the database indicates that no meter multiplier was supplied.

**Default:**

None, 0 stored in the database if not supplied.

**Note:** If 0 is stored and/or observed in the database and a cut is exported to the LSE format, all meter fields for that cut will be absent. A zero stored in the database is indicative of “no meter information supplied”.

**Meter Offset**

**Relationships:**

Must be supplied if any of the other Meter fields (Meter Start Reading, Meter Stop Reading, or Meter Multiplier) are supplied.

**Requirements:**

If supplied, value must be numeric.

**Default:**

0 stored in the database if not supplied

**Pulse Multiplier**

**Relationships:**

None.

**Requirements:**

Value must be positive numeric. Also, must be supplied if Pulse Offset (see below) is supplied.

**Default:**

None, 0 stored in the database if not supplied.

**Note:** If 0 is stored and/or observed in the database and a cut is exported to the LSE format, all pulse fields for that cut will be absent. A zero stored in the database is indicative of “no pulse information supplied”.

## Pulse Offset

### Relationships:

If supplied, you must also supply Pulse Multiplier.

### Requirements:

Value must be numeric.

### Default:

None. (Zero is stored in the database to indicate that the Pulse Multiplier was not supplied.)

## Seconds Per Interval (SPI)

### Description:

This field stores the interval duration if appropriate. An Intervals Per Hour (IPH) of 1 can be translated to a Seconds Per Interval of 3600.

### Relationships:

Value of 0 indicates non-uniform recording duration and a full array of interval start values (one per recording) located in the data section. (This feature will be supported in a future release and is **not currently available**.)

### Requirements:

- Value must be a positive integer.
- Valid values for SPI are 86400, 3600, 1800, 900, 300, and 60.

### Default:

None; field is required.

## Unit of Measure

### Description:

See **Units of Measure** on page C-16 for valid Units of Measure.

### Relationships:

None.

### Requirements:

- Value should be a valid unit of measure. If the value entered is unknown to, 16 (Miscellaneous) will be stored.
- Value must be non-negative numeric.
- Maximum value is 32767.

### Default:

None; field is required.

## Basic Unit Code

### Description:

This field is reserved for future use.

### Relationships:

None.

### Requirements:

Omit this field.

### Default:

0.

## Times Zones West of GMT

### Description:

This value tells programs accessing this record which time zone the values have been recorded in. Its value will be the difference in time, heading west, between Greenwich Mean Time and the represented time zone as measured in half hours. Eastern Standard Time should have a value of 10; Pacific = 16. A Value of -1 will indicate "Time Zone not available".

### Relationships:

None.

### Requirements:

Value must be -1 or a non-negative numeric between 0 and 47, that represents the time zones West of GMT measured in half-hour increments. Any record containing a time zone greater than 47 will be rejected.

### Default:

-1.

## Population

### Description:

For Oracle Utilities Load Analysis statistical records only.

### Relationships:

None.

### Requirements:

Value may be any non-negative integer value.

### Default:

0.

## Weight

### Description:

For Oracle Utilities Load Analysis statistical records only.

### Relationships:

None.

### Requirements:

Value may be any non-negative numeric value.

### Default:

0.0.

## Time Zone Standard Name

### Description:

Defines the Time Zone to which the cut is associated. The value must be one of EST, CST, MST, PST, or be defined in the LSCALENDAR.XML configuration file (if present). Each value in this field must contain printable ASCII characters no longer than 32 bytes.

Only letters, numbers, underscores, and hyphens will be accepted in this field, all other characters and embedded blanks will cause the record to be rejected.

### Relationships:

Must map exactly to one of EST, CST, MST, PST or an entry in the LSCALENDAR.XML file (if present). Oracle Utilities Load Analysis will import without checking this file.

### Requirements

Length: <= 32 bytes

## Descriptor

### Description:

The user may enter any descriptive information in this field. Any information entered into this field will be stored in the database verbatim as supplied.

### Relationships:

None.

### Requirements:

- Value must be 80 characters or fewer, and may contain commas.
- May start with blanks.
- If more than 80 characters are entered, truncation will occur and the first 80 will be used.

## Timestamp

### Relationships:

None.

### Requirements:

- Format should be YYYYMMDDHHMMSSmmm. The milliseconds (mmm) may be omitted, in which case mmm will be set to 000.
- No spaces, colons, or any ASCII value allowed between time components.

### Default:

Current time that data is loaded into database.

**Note:** It is strongly recommended that you leave this field empty and allow the system to input the TimeStamp for you.

## Origin

### Relationships:

None.

### Requirements:

Must be one of: M (metered), P (profiled), C (computed), or S (Oracle Utilities Load Analysis Statistic).

### Default:

M (metered).

## Interval Value

### Description:

This field contains the actual recorded value for a time period.

### Relationships:

Must have one status code entry per interval data value.

### Requirements:

Any numeric value. (Up to 15 significant digits can be stored.)

### Default:

0 (see Status Code, Relationships, below).

## Status Code

### Description:

This field contains the status code for the preceding Interval Value.

### Relationships:

Must have an entry for the Interval Value.

### Requirements:

Any character will be accepted.

### Default:

- This field defaults to a blank ( ' ') when omitted if the interval value is present.
- This field defaults to '9' when the Interval Value field is omitted.

## Interval Start

### Description:

- This field indicates the starting time of the recording or the time when the recording took place.
- Format must be YYYYMMDDHHMMSS. (This feature will be supported in a future release and is **not currently available**.)

### Relationships:

- Until this feature is supported, the SPI may not be omitted (represented by „).).
- There must be one entry per Interval Value.

### Requirements:

See Start Time, Requirements.

### Default:

None.

## Enhanced Format Details

### First Header Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	Must be 00000001
2	Recorder/ Customer Identifier	64	Letters, numbers, hyphens or underscores are acceptable Values
3	Channel	5	Max is 32767
4	Start Time	14	YYYYMMDDHHMMSS (24-hour)
5	Stop Time	14	YYYYMMDDHHMMSS (24-hour)
6	DST Participant Flag	1	Y/N/A
7	Invalid Record Flag	1	Y/N

### Second Header Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	Must be 00000002 Default: None; field required
2	Meter Start Reading	N/A	Non-Negative Numeric Max: 999999999999999.9999 Default: 0
3	Meter Stop Reading	N/A	Non-Negative Numeric Max: 999999999999999.9999 Default: 0
4	Meter Multiplier	N/A	Positive Numeric (Optional) Max: 999999999999999.9999 Default: 0
5	Meter Offset	N/A	Numeric (Optional) Max: 999999999999999.9999 Default: 0
6	Pulse Multiplier	N/A	Positive Numeric (Optional) Max: 999999999999999.9999 Default: 0
7	Pulse Offset	N/A	Numeric (Optional) Max: 999999999999999.9999 Default: None; field required
8	Seconds Per Interval (SPI)	N/A	Positive Numeric Default: None; field required

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
9	Unit of Measure	N/A	Numeric Max: 32767 Default: None; field required
10	Basic Unit Code	N/A	Positive Numeric (Optional) Max: 9999 Default: 0
11	Time Zones West of GMT	N/A	Numeric. (Optional) Default: -1 Min: -1 Max: 47
12	Population	N/A	Positive Numeric (Optional) Max: 999999999999999.9999 Default: 0.0
13	Weight	N/A	Positive Numeric (Optional) Max: 999999999999999.9999 Default: 0.0
14	Time Zone Standard Name		Character (32) Overrides any value in TZWGMT. Must be one of "EST", "CST", "MST", "PST", or definition must exist in LSCalendar.xml file.

### Third Header Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	Must be 00000003
2	Descriptor	80	

### Fourth Header Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	Must be 00000004
2	Timestamp	17	YYYYMMDDHHMMSSMM (optional) Default: current date
3	Origin	1	C, M, P, or S (optional)

Data Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	10000000 through 99999999
2	Interval Value		
3	Status Code	1	
4	Interval Start	14	YYYYMMDDHHMMSS (optional)

A data line may contain as many full data-status-time groups as desired, providing the record does not exceed a character count (record length) of 32750.

Interval start times may be omitted, but a comma must still be present as a placeholder. Data must be contiguous and “missings” (or gaps) must be represented by value 0 and status '9'. At this time, all intervals must be of the same duration, hence Interval Start Times should not be specified.

## Sample Files

### Enhanced Direct Input File

A sample input file using the enhanced format with 3 headers and 2 data records and with interval starts omitted:

```
00000001,N1732,17,19820412000001,19820412005959,Y,Y
00000002,23887.34,23903.56,1.0,0.0,0.0144,0.0,900,1,0,-1,,
00000003,ACCEPTANCE TEST DATA NUMBER 1
00000004,,M
10000000,31,Q,,9.52,R,,.00314,S,,123456.98765,T,,
10000001,32,Q,,9.53,R,,.00315,S,,123457,T,,
```

### Standard Direct Input File

A sample old-format file is provided below for comparison.

```
0001WARNINGTEST 10412821351051382140004010
0002013127501400250000000000912350000000000945679-000000000012346-
0000000000034567
0003ACCEPTANCE TEST DATA 00000000000000000000000000000000
0004
100000031Q00028Q00008Q00061Q00038Q00011Q00050Q00005Q00055Q00000900000900004Q
100100029Q00016Q00079Q00036Q00020Q00076600061600015600078Q00021Q00050Q00004Q
100200004Q00007Q00006Q00006Q00004Q00008Q00005Q00009Q00009Q00009Q00005Q00006Q
```

## Units of Measure

The table below lists valid Units of Measure. Intervals may be summed or averaged when aggregating, depending on the unit of measure. The aggregation technique is noted on the table

**Note:** Oracle Utilities products also accept 3-digit UOMs, which may be user-defined.

CODE	DESCRIPTION	AGGREGATION TECHNIQUE
01	kWh	Sum
02	kW	Average
03	kVARh	Sum
04	kVAH	Sum
05	Temperature (°F)	Average
06	KQD	Average
07	V <sup>2</sup> H (PTP)	Sum
08	kQh	Sum
09	kQh (45 degrees)	Sum
10	I <sup>2</sup> H	Sum
11	Volts	Average
12	Amps	Average
13	Temperature (°C)	Average
14	Dewpoint	Average
15	Amplitude	Average
16	Miscellaneous	Sum
17	Minute Run Time (MRT)	Sum
18	Wind Velocity (Cm/Sec)	Average
19	Fraction V2H (PTN)	Average
20	Percent	Average
21	Flow	Average
22	kVAR	Average
23	kVA	Average
24	kVA Ratio	Average
25	Power Factor	Average
26	Hertz	Average
27	Feet	Average

CODE	DESCRIPTION	AGGREGATION TECHNIQUE
28	Minutes	Sum
29	On / Off (Tap position)	Average
30	Inches	Average
31	Individual kWh	Sum
32	kWh r	Sum
33	Individual Totalized kVARh	Sum
34	kVARh r	Sum
35	Individual Totalized Temperature (°F)	Average
36	kVAh r	Sum
37	IND V <sup>2</sup> H (Individual totalized V <sup>2</sup> H)	Sum
38	IND kQh (Individual totalized kQh)	Sum
39	KQH r	Sum
40	Miscellaneous	Average
41	IND Volts (Individual totalized Volts)	Average
42	IND Amps (Individual totalized Amperes)	Average
43	IND Temperature (°C) (Individual totalized temperature, degrees Celsius)	Average
44	Mw	Sum
45	MVAR	Average
46	MVA	Average
47	IND MRT (Individual totalized MRT)	Sum
48	IND CMS (Individual totalized CMS)	Average
49	Run Hours	Sum
50	Equivalent Full Load Hours	Sum
51	KWH-OUT	Sum
52	KW-OUT	Average
53	KVARH-OUT	Sum
54	KVAH-OUT	Sum
55	KQH-OUT	Sum
56	LEAD-KVARH	Sum
57	LEAD-KVARH-OUT	Sum
58	LAG-KVARH	Sum

CODE	DESCRIPTION	AGGREGATION TECHNIQUE
59	LAG-KVARH-OUT	Sum
60	Gallons / Minute	Average
61	BTU	Sum
62	THERMS	Sum
63	Cubic Feet / Minute	Average
64	Cubic Feet / Second	Average
65	WM <sup>2</sup>	Average
66	Relative Humidity	Average
67	MPH	Average
68	THI	Average
69	Gallons	Sum
70	Cubic Feet	Sum
71	Temperature Difference	Average
72	KVAR-OUT	Average
73	KVA-OUT	Average
74	Knots	Average
75	Degrees	Average
76	CCF (Hundred cubic feet)	Sum
77	CF / Hour	Average
78	Pounds / Square inch	Average
79	Dollars	Sum
80	DECATHERMS	Sum
81	Pounds	Sum
82	Pounds / Hour	Average
83	MPOUNDS	Sum
84	MPOUNDS / Hour	Average
85	\$/KWH	Average
86	\$/MW	Average
87	\$/MWH	Average
88	\$/Hour	Average
89	Volt Hours	Sum
90	Individual Totalized Cubic Feet	Sum

CODE	DESCRIPTION	AGGREGATION TECHNIQUE
91	Individual Totalized Btu	Sum
92	Pressure in MILLIBARS	Average
93	Visibility in Miles	Average
94	Cents per kWh	Average
99	Individual Totalized Gallons	Sum
100	MWH	Sum
102	Euros	Sum
103	Euros per MWH	Average
104	Euros MW	Average
105	GW	Average
106	TWH	Sum
107	Cubic Meters (M3)	Sum
108	Mega Joules per Cubic Meter (MJ/m3)	Average
109	Kilograms per Cubic Meter (Kg/m3)	Sum
110	Cubic Meters per Hour (M3/H)	Average

### UOM Compatibility

When merging cuts with different units-of-measure, cuts with specific differing UOMs can be combined while others can't. For example, you can't merge two cuts if the UOM of one cut is inches and the UOM of the other is dollars, because whichever UOM is assigned to the merged cut wouldn't apply to at least some of the data.

However, cuts with certain specific different UOMs can be combined. UOMs that can be combined are referred to as compatible UOMs, and are listed in the table below.

Compatible UOMs
01,31,32,51
02,52
03,33,34,53,56,57,58,59
04,36,54
05,35
07,37
08,09,38,39,55
11,41
12,42
13,43

Compatible UOMs
17,47
18,48
22,72
23,73
61,69
69,99
70,90

These units of measures may have their intervals divided by the IPH for display:

Demand-Type UOMs
02, 05, 06, 07, 10, 22, 23, 24, 52, 72, 73, 105, 110

# Appendix D

---

## Oracle Utilities Standard XML Interval Data Format

This appendix provides a detailed description of the Oracle Utilities Standard XML Interval Data Formats, supported by Oracle Utilities applications. This includes:

- **Standard XML Format**
- **XML Standard File Format Example**
- **XML Standard File Format DTD**
- **XML Compact Format**

# Standard XML Format

## General Node Descriptions

In the Oracle Utilities Standard XML format, nodes are enclosed with XML tags. A data line may contain as many full data-status-time groups as desired.

Numeric nodes will be expressed as decimal numbers, and may contain a decimal point and/or a leading minus sign (-). Non-negative numbers are assumed on all numeric nodes if no leading minus sign is present. If a minus sign is present, spaces may exist between it and the numeric value. For example: ..., - 122.11,... or ..., -23.3,... are valid entries for a numeric node.

The Recorder, Channel Number, and the Start Time compose the full key of the record which, in combination with each other, must uniquely identify the record to be stored in the database.

Node names and XML tag names must be uppercase.

Any cut entered with an invalid Start or Stop Time will be rejected.

The contents of the Recorder node, Origin node, and all Y/N indicators will be translated to uppercase before they are stored in the database.

**Note:** Database sort order will follow ASCII collating sequence, starting with the Recorder and followed in turn by the Channel, then the Start Time.

A cut will be rejected if the number of interval values supplied is greater than the number of interval values expected, based upon the Start Time, Stop Time, SPI, and DST\_Participant nodes. If too few interval values are supplied, the cut will be filled out with “missing” (value of 0 and status of ‘9’).

Any Stop Time falling exactly on an interval boundary will have one second subtracted from it.

## General Rules

Each root node of a file starts with a set of special nodes. These nodes contain the name of the format, the version number of the format, the decimal point specification, and date/time format specification.

### Format Name

The first node identifies the format as “LODESTAR Interval Data XML Format”. It is an error if this is not present.

#### Example:

```
<INTERVAL_DATA_FORMAT>LODESTAR Interval Data XML Format </INTERVAL_DATA_FORMAT>
```

### Format Version Number

The second node identifies the version number of the format. It is an error if this is not present.

#### Example:

```
<VERSION>version_number</VERSION>
```

### Decimal Point Specification

The third node identifies the decimal point specification to be used in the file. All float values used throughout the file must use this decimal point format. It is an error if this is not present.

#### Example:

```
<DECIMAL_POINT>.</DECIMAL_POINT>
```

### Date/Time Format Specification

The fourth node identifies the date/time format to be used in the file. There are three available date/time formats:

1. MM/DD/YYYY HH:MM:SS
2. DD/MM/YYYY HH:MM:SS
3. YYYY/MM/DD HH:MM:SS

All date/time values used throughout the file must use this format. It is an error if this is not present.

#### Example:

```
<DATETIME_FORMAT>10/27/1999 10:31:00</DATETIME_FORMAT>
```

These nodes are followed by the interval cut information as outlined below:

## Node Relationships and Requirements

Relationships between nodes exist in the standard XML format. These are described in this section, and detailed node requirements are provided. Defaults are inserted when the processing program encounters an omitted node.

### RECORDER (key element)

**Relationships:**

None.

**Requirements:**

- Only letters, numbers, underscores, and hyphens will be accepted in this node; all other characters and embedded blanks will cause the record to be rejected.

**Note:** Lowercase letters will be translated to uppercase before they are stored in the database.

- Must be at least one character, and no more than 64 characters in length.

**Default:**

None; node is required.

### CHANNEL (key element)

**Relationships:**

None.

**Requirements:**

- Node values must be non-negative integer only.
- Maximum value is 32767.

**Default:**

None; node is required.

### STARTTIME (key element)

**Relationships:**

- Must be a valid datetime value less than the value of the Stop Time.

**Requirements:**

- Format must match that specified by the **Date/Time Format Specification** on page D-3.
- Value must be after 19700101000000.
- No spaces, colons, or any ASCII value allowed between time components.

**Default:**

None; node is required.

## STOPTIME

### Relationships:

- Must be a valid datetime value greater than the value of the Start Time.
- Value must agree with the number of interval values past the start time \* SPI.
- One second is subtracted if the value is equal to an even interval start time.

### Requirements:

See STARTTIME.

### Default:

None; node is required.

## DST\_PARTICIPANT

### Relationships:

None.

### Requirements:

Valid values are Y, N, or A, indicating whether or not this record will be processed using DST adjustments.

Flag Value	Definition
Y	(DST Participant) The intervals recorded participate in DST adjustments and the intervals during DST reflect the time change.
N	(Does not participate in DST) The intervals recorded do not make any DST time changes.
A	(DST participant) The intervals have been adjusted to 24-hour days. (At the April time change, there should be a value at the time change (a place holder) of 0 with status code "9". At the October time change, there should be a combined value.) These cuts will be converted to a "Y" or "N", and are not stored in the database as "A".

### Default:

As specified in the INTDDEFAULTDST configuration parameter setting, the INTDCONFIG.XML file, or the CSDST value in Oracle Utilities Load Analysis.

## VALIDATION\_REQUIRED

**Relationships:**

None.

**Requirements:**

Valid values are Y or N indicating whether or not the record should be marked as valid.

Flag Value	Definition
Y	The incoming record contains unvalidated data.
N	The incoming record contains valid data, and should be marked as such.

**Default:**

Y.

## METER\_START\_READING

**Relationships:**

Must be supplied if any of the other Meter nodes (Meter Stop Reading, Meter Multiplier, or Meter Offset) are supplied.

**Requirements:**

If supplied, value must be a non-negative numeric value.

**Default:**

0.

## METER\_STOP\_READING

**Relationships:**

Must be supplied if any of the other Meter nodes (Meter Start Reading, Meter Multiplier, or Meter Offset) are supplied.

**Requirements:**

If supplied, value must be a non-negative numeric value.

**Default:**

0 (not supplied).

## METER\_MULTIPLIER

**Relationships:**

Must be supplied if any of the other Meter nodes (Meter Start Reading, Meter Stop Reading, or Meter Offset) are supplied.

**Requirements:**

If supplied, value must be positive numeric. However, if no value is specified, 0 will be stored. Zero (0) is not a valid value for the meter multiplier, but its presence in the database indicates that no meter multiplier was supplied.

**Default:**

None, 0 stored in the database if not supplied.

**Note:** If 0 is stored and/or observed in the database and a cut is exported to the LSE format, all meter fields for that cut will be absent. A zero stored in the database is indicative of “no meter information supplied”.

**METER\_OFFSET****Relationships:**

Must be supplied if any of the other Meter nodes (Meter Start Reading, Meter Stop Reading, or Meter Multiplier) are supplied.

**Requirements:**

If supplied, value must be numeric.

**Default:**

0 (not supplied).

**PULSE\_MULTIPLIER****Relationships:**

None.

**Requirements:**

Value must be positive numeric. Also, must be supplied if Pulse Offset (see below) is supplied.

**Default:**

None, 0 stored in the database if not supplied.

**Note:** If 0 is stored and/or observed in the database and a cut is exported to the LSE format, all pulse fields for that cut will be absent. A zero stored in the database is indicative of “no pulse information supplied”.

**PULSE\_OFFSET****Relationships:**

If supplied, you must also supply Pulse Multiplier.

**Requirements:**

Value must be numeric.

**Default:**

None.

**SPI (Seconds per Interval)****Description:**

This node will store the interval duration if appropriate. An Intervals per Hour (IPH) of 1 can be translated to an SPI of 3600.

**Relationships:**

Value of 0 indicates non-uniform recording duration and a full array of interval start values (one per recording) located in the data section. (This feature will be supported in a future release and is **not currently available**.)

**Requirements:**

- Value must be a positive integer.
- Valid values for SPI are 86400, 3600, 1800, 900, 300, and 60.

**Default:**

None; node is required.

**UOM (Unit of Measure)****Description:**

See **Units of Measure** on page B-16 for valid Units of Measure.

**Relationships:**

None.

**Requirements:**

- Value should be a valid unit of measure. If the value entered is unknown to, 16 (Miscellaneous) will be stored.
- Value must be non-negative numeric.
- Maximum value is 32767.

**Default:**

None; node is required.

**DC\_FLOW****Description:**

Power flow direction. This is used with MV90 data.

**Relationships:**

None.

**Requirements:**

Must be either “D” (Delivered) or “R” (Received).

**Default:**

“D”

**TIMEZONE****Description:**

This value tells programs accessing this record which time zone the values were recorded in. Its value will be the difference in time, heading west, between Greenwich Mean Time and the represented time zone, measured in half hours. Eastern Standard Time should have a value of 10; Pacific = 16. A Value of -1 indicates “Time Zone not available.”

**Relationships:**

None.

**Requirements:**

Value must be -1 or a non-negative numeric between 0 and 47, that represents the time zones west of GMT, measured in half-hour increments. Any record containing a time zone greater than 47 will be rejected.

**Default:**

-1.

**POPULATION****Description:**

For Oracle Utilities Load Analysis Statistical records only.

**Relationships:**

None.

**Requirements:**

Value may be any non-negative integer value.

**Default:**

0.

**WEIGHT****Description:**

For Oracle Utilities Load Analysis Statistical records only.

**Relationships:**

None.

**Requirements:**

Value may be any non-negative numeric value.

**Default:**

0.0.

## TIME\_ZONE\_STANDARD\_NAME

### Description:

Defines the Time Zone to which the cut is associated. The value must be one of EST, CST, MST, PST, or be defined in the LSCALENDAR.XML configuration file. Each value in this field must contain printable ASCII characters no longer than 32 bytes.

Only letters, numbers, underscores, and hyphens will be accepted in this field, all other characters and embedded blanks will cause the record to be rejected.

**Note:** Lowercase letters will be translated to uppercase before being stored

### Relationships:

Must map exactly to one of EST, CST, MST, PST or an entry in the LSCALENDAR.XML file. Oracle Utilities Load Analysis will import without checking this file.

### Requirements

Length: <= 32 bytes

## DESCRIPTOR

### Description:

The user may enter any descriptive information in this node. Any information entered into this node will be stored in the database verbatim.

### Relationships:

None.

### Requirements:

- Value must be 80 characters or fewer, and may contain commas.
- May start with blanks.
- If more than 80 characters are entered, truncation occurs and the first 80 are used.

## TIMESTAMP

### Relationships:

None.

### Requirements:

- Format must match that specified by the **Date/Time Format Specification** on page D-3.
- No spaces, colons, or any ASCII value allowed between time components.

### Default:

Current time that data is loaded into database.

**Note:** It is strongly recommended that you leave this node empty and allow the system to input the TimeStamp for you.

## ORIGIN

### Relationships:

None.

### Requirements:

Must be one of: M (metered), P (profiled), C (computed), or S (Oracle Utilities Load Analysis Statistic).

**Default:**

M (metered).

**ACCEPT\_REJECT\_STATUS****Description:**

This is used by MV90 to indicate if the cut was accepted or rejected.

**Relationships:**

None.

**Requirements:**

Will be either:

Status Code	Description
00	Accept. No Error.
01 though 07	Rejected/Reject type.

**Default:**

None.

**TRANSLATION DATE****Description:**

This is used by MV90 to indicate the date/time the cut was translated from MV90.

**Relationships:**

None.

**Requirements:**

Format must match that specified by the **Date/Time Format Specification** on page D-3).

**Default:**

None.

## Interval Values

The interval values data is enclosed within the <INTERVAL> node. Each individual interval value is enclosed within the <RECORD> node, and consists of the VALUE, STATUS (optional), and STARTTIME (optional). The RECORD node will not have a STATUS node if and only if the interval status is a space. The optional STARTTIME node in the RECORDING node is only included if the XMLEXPORTINTDSTARTTIME=1 parameter is included in the LODESTAR.CFG file (see **Chapter Four: Configuration Files** in the *Oracle Utilities Energy Information Platform Installation and Configuration Guide*).

Descriptions of the RECORD nodes are as follows:

### VALUE (Interval Value)

**Description:**

This node contains the actual recorded value for a time period.

**Relationships:**

Must have one status code entry per interval data value, if the status is not a space (see above).

**Requirements:**

Any numeric value. (Up to 15 significant digits can be stored.)

**Default:**

0 (see Status Code, Relationships, below).

### STATUS (Status Code)

**Description:**

This node contains the status code for the preceding Interval Value.

**Relationships:**

Must have an entry for the Interval Value.

**Requirements:**

Any character will be accepted.

**Default:**

- This node defaults to a blank (‘ ’) when omitted if the interval value is present.
- This node defaults to ‘9’ when the Interval Value node is omitted.

## STARTTIME (Interval Start Time)

### Description:

- This node indicates the starting time of the recording or the time when the recording took place.
- Format must match that specified by the **Date/Time Format Specification** (page D-3).
- This feature will be supported in a future release and is **not currently available**.

### Relationships:

- There can be one entry per Interval Value.

### Requirements:

See Start Time, Requirements.

### Default:

None.

# XML Standard File Format Example

An example of the Oracle Utilities Standard XML format is shown below.

```
<INTERVAL_DATA>
 <INTERVAL_DATA_FORMAT>LODESTAR Interval Data XML Format</INTERVAL_DATA_FORMAT>
 <VERSION>1.2</VERSION>
 <DECIMAL_POINT>.</DECIMAL_POINT>
 <DATETIME_FORMAT>MM/dd/yyyy HH:mm:ss</DATETIME_FORMAT>
 <CUT>
 <RECORDER>TV011285</RECORDER>
 <CHANNEL>32767</CHANNEL>
 <STARTTIME>01/01/1999 00:00:00</STARTTIME>
 <STOPTIME>01/31/1999 23:59:59</STOPTIME>
 <DST_PARTICIPANT>Y</DST_PARTICIPANT>
 <VALIDATION_REQUIRED>N</VALIDATION_REQUIRED>
 <METER_START_READING>1234</METER_START_READING>
 <METER_STOP_READING>2345</METER_STOP_READING>
 <METER_MULTIPLIER>1.12345</METER_MULTIPLIER>
 <METER_OFFSET>100.0</METER_OFFSET>
 <PULSE_MULTIPLIER>1.0000</PULSE_MULTIPLIER>
 <PULSE_OFFSET>0.0000</PULSE_OFFSET>
 <SPI>900</SPI>
 <UOM>01</UOM>
 <DC_FLOW>D</DC_FLOW>
 <TIMEZONE>8</TIMEZONE>
 <POPULATION>300000</POPULATION>
 <WEIGHT>0.0</WEIGHT>
 <TIME_ZONE_STANDARD_NAME>EST</TIME_ZONE_STANDARD_NAME>
 <DESCRIPTOR>DEMO</DESCRIPTOR>
 <TIMESTAMP>01/01/2000 00:00:00</TIMESTAMP>
 <ORIGIN>M</ORIGIN>
 <INTERVAL>
 <RECORDING>
 <VALUE>123456.1234</VALUE>
 <STATUS>A</STATUS>
 <STARTTIME>11/01/1999 11:12:34</STARTTIME>
 </RECORDING>
 <RECORDING>
 <VALUE>123456.1234</VALUE>
 <STATUS>A</STATUS>
 <STARTTIME>11/01/1999 11:15:00</STARTTIME>
 </RECORDING>
 </INTERVAL>
 </CUT>
</INTERVAL_DATA>
```

# XML Standard File Format DTD

The Data Type Definition for the Oracle Utilities Standard XML format is shown below.

```
<!DOCTYPE INTERVAL_DATA
[
<!ELEMENT INTERVAL_DATA (INTERVAL_DATA_FORMAT, VERSION, DECIMAL_POINT,
 DATETIME_FORMAT, INTERVAL_CUT+)>
<!ELEMENT INTERVAL_DATA_FORMAT (#PCDATA)>
<!ELEMENT VERSION (#PCDATA)>
<!ELEMENT DECIMAL_POINT (#PCDATA)>
<!ELEMENT DATETIME_FORMAT (#PCDATA)>
<!ELEMENT CUT (RECORDER, CHANNEL, STARTTIME, STOPTIME, DST_PARTICIPANT,
 METER_START_READING, METER_STOP_READING, METER_MULTIPLIER,
 METER_OFFSET, PULSE_MULTIPLIER, PULSE_OFFSET, SPI, UOM, TIMEZONE,
 TIMESTAMP, ORIGIN?, DC_FLOW?, POPULATION?, WEIGHT?,
 TIME_ZONE_STANDARD_NAME?, DESCRIPTOR?, ACCEPT_REJECT_STATUS?,
 TRANSLATION_DATE?, INTERVAL)>
<!ELEMENT RECORDER (#PCDATA)>
<!ELEMENT CHANNEL (#PCDATA)>
<!ELEMENT STARTTIME (#PCDATA)>
<!ELEMENT STOPTIME (#PCDATA)>
<!ELEMENT DST_PARTICIPANT (#PCDATA)>
<!ELEMENT METER_START_READING (#PCDATA)>
<!ELEMENT METER_STOP_READING (#PCDATA)>
<!ELEMENT METER_MULTIPLIER (#PCDATA)>
<!ELEMENT METER_OFFSET (#PCDATA)>
<!ELEMENT PULSE_MULTIPLIER (#PCDATA)>
<!ELEMENT PULSE_OFFSET (#PCDATA)>
<!ELEMENT SPI (#PCDATA)>
<!ELEMENT UOM (#PCDATA)>
<!ELEMENT DC_FLOW (#PCDATA)>
<!ELEMENT TIMEZONE (#PCDATA)>
<!ELEMENT POPULATION (#PCDATA)>
<!ELEMENT WEIGHT (#PCDATA)>
<!ELEMENT TIME_ZONE_STANDARD_NAME (#PCDATA)>
<!ELEMENT DESCRIPTOR (#PCDATA)>
<!ELEMENT TIMESTAMP (#PCDATA)>
<!ELEMENT ORIGIN (#PCDATA)>
<!ELEMENT ACCEPT_REJECT_STATUS (#PCDATA)>
<!ELEMENT TRANSLATION_DATE (#PCDATA)>
<!ELEMENT INTERVAL (RECORDING+)>
<!ELEMENT RECORDING (VALUE, STATUS?, STARTTIME?)>
<!ELEMENT VALUE (#PCDATA)>
<!ELEMENT STATUS (#PCDATA)>
<!ELEMENT STARTTIME (#PCDATA)>
]>
```

## XML Compact Format

The XML “compact” format is an XML format used with interval data that has a smaller footprint than the XML standard format. The compact format is the same format used by the LSINTD Interval Data component, and can be used with the Oracle Utilities Adapter. See the *Oracle Utilities Energy Information Platform User's Guide* and *Oracle Utilities Energy Information Platform Installation and Configuration Guide* for more information about the *Oracle Utilities* Adapter.

### XML Compact File Format Examples

The following is an example of a standard interval data cut defined in the XML compact format.

```
<CUTS>
 <CUTEX>
 <HEADER
 RECORDER="1700"
 CHANNEL="1"
 STARTTIME="2003-02-24 00:00:00"
 STOPTIME="2003-02-24 23:59:59"
 START_READ=15000
 STOP_READ=17500
 M_MULT=1
 M_OFFSET=0
 SPI="1800"
 UOM="1"
 DESC="Pre-validation"]
 DCFLOW="D"
 ARS="00"
 TZSN="EST"
 ORIGIN="M"
 DST="Y"
 CHNSTATUS="0" />
 <INTS>
 <I V=12.00 S=" " E=" " />
 <I V=13.00 S=" " E=" " />
 <I V=14.00 S=" " E=" " />
 ...
 </INTS>
 </CUTEX>
</CUTS>
```

Below is an example the compact XML format user to import data into an enhanced/generic interval data table that includes the CALCGROUP, MINIMUM, and MAXIMUM custom columns.

```
<CUTS>
 <CUTEX>
 <HEADER TIMESERIES="TEST,1" SPI="900" UOM="1" DST="Y" STARTTIME="2006-12-12
00:01:00" STOPTIME="2007-01-10 23:59:59" TDATE="1969-12-31 19:00:00" ORIGIN="C"
DCFLOW="" ARS="" START_READ="0.0" STOP_READ="0.0" POP="0.0" WEIGHT="0.0"
M_OFFSET="0.0" M_MULT="0.0">
 <CUSTOM CALCGROUP="1" MAXIMUM="10" MINIMUM="-10" />
 </HEADER>
 <INTS>
 <I V="16" S="7" />
 <I V="18" S="7" />
 <I V="16" S="7" />
 <I V="18" S="7" />
 <I V="15" S="7" />
 <I V="18" S="7" />
 <I V="16" S="7" />

 <I V="21" S="7" />
 <I V="20" S="7" />
 <I V="17" S="7" />
```

```

 </INTS>
 </CUTEX>
</CUTS>

```

Below is a similar example that includes a category.

```

<CUTS>
 <CUTEX>
 <HEADER TIMESERIES="PERCOT,1" CATEGORY="RAW" UOMCODE="01" STARTTIME="2005-
12-03T00:00:00" STOPTIME="2005-12-03T23:59:59" SPI="3600" DST="Y" P_MULT="1.0"
P_OFFSET="1.0">
 <CUSTOM CALCGROUP="1" MAXIMUM="1" MINIMUM="1"/>
 </HEADER>
 <INTS>
 <I V="2850" S=" " />
 <I V="3150" S=" " />
 <I V="2850" S=" " />
 <I V="3150" S=" " />
 <I V="2850" S=" " />
 <I V="3150" S=" " />
 <I V="2850" S=" " />
 <I V="3150" S=" " />
 <I V="2850" S=" " />
 <I V="3150" S=" " />
 <I V="2850" S=" " />
 <I V="3150" S=" " />
 <I V="2850" S=" " />
 <I V="3150" S=" " />
 <I V="2850" S=" " />
 <I V="3150" S=" " />
 </INTS>
 </CUTEX>
</CUTS>

```

## XML Compact File Format Element and Attribute Descriptions

The elements and attributes of the XML compact format are described below. All attributes correspond to elements described under **Standard XML Format** on page D-2. Defaults are inserted when the processing program encounters an omitted attribute.

**CUTS:** Optional element containing one or more CUTEX elements.

**CUTEX:** Element containing a single interval data cut.

Elements:

**HEADER:** Element containing header information for the interval data cut contained in the parent CUT element.

**Attributes:**

- **RECORDER** (key attribute): Recorder ID for the interval data cut. See **RECORDER (key element)** on page D-4 for detailed information.
- **CHANNEL** (key attribute): Channel number for the interval data cut. See **CHANNEL (key element)** on page D-4 for detailed information.
- **TIMESERIES** (key attribute): Identity of the parent record of the interval data. Used in place of the RECORDER and CHANNEL elements when defining interval data stored in enhanced interval data tables.
- **CATEGORY:** Optional category of the interval data. Used only with interval data stored in enhanced interval data tables..
- **STARTTIME** (key attribute): Start time for the interval data cut. Must be in ISO datetime format. See **STARTTIME (key element)** on page D-4 for detailed information.
- **STOPTIME:** Stop time for the interval data cut. Must be in ISO datetime format. See **STOPTIME** on page D-5 for detailed information.

- **TIMESTAMP**: (optional) Timestamp of the interval data cut. Must be in ISO datetime format. See **TIMESTAMP** on page D-10 for detailed information.
- **START\_READ**: (optional) Meter start reading for the interval data cut. See **METER\_START\_READING** on page D-6 for detailed information.
- **STOP\_READ**: (optional) Meter stop reading for the interval data cut. See **METER\_STOP\_READING** on page D-6 for detailed information.
- **M\_MULT**: (optional) Meter multiplier for the interval data cut. See **METER\_MULTIPLIER** on page D-6 for detailed information.
- **M\_OFFSET**: (optional) Meter offset for the interval data cut. See **METER\_OFFSET** on page D-7 for detailed information.
- **POP**: (optional) Population for the interval data cut. See **POPULATION** on page D-9 for detailed information.
- **WEIGHT**: (optional) Weight for the interval data cut. See **WEIGHT** on page D-9 for detailed information.
- **SPI** (Seconds per Interval): SPI for the interval data cut. See **SPI (Seconds per Interval)** on page D-7 for detailed information.
- **UOM** (Unit of Measure): UOM for the interval data cut. See **UOM (Unit of Measure)** on page D-8 for detailed information.
- **DESC**: (optional) Descriptor for the interval data cut. See **DESCRIPTOR** on page D-10 for detailed information.
- **DCFLOW**: (optional) DC Flow for the interval data cut. See **DC\_FLOW** on page D-8 for detailed information.
- **ARS**: (optional) Accept-Reject-Status for the interval data cut. See **ACCEPT\_REJECT\_STATUS** on page D-11 for detailed information.
- **TDATE**: (optional) Translation Date for the interval data cut. See **TRANSLATION DATE** on page D-11 for detailed information.
- **TZSN**: (optional) Time Zone Standard Name for the interval data cut. See **TIME\_ZONE\_STANDARD\_NAME** on page D-10 for detailed information.
- **ORIGIN**: (optional) Origin for the interval data cut. See **ORIGIN** on page D-10 for detailed information.
- **DST**: (optional) DST participant flag for the interval data cut. See **DST\_PARTICIPANT** on page D-5 for detailed information.
- **CHNSTATUS**: (optional) Status code for the interval data cut.

**Elements:**

- **CUSTOM**: Used to define custom columns added to enhanced/generic interval data tables. The format of this element is:

```
<CUSTOM <fieldname>=<value>/>
```

where:

- **<fieldname>** is an attribute that is the name of the custom column
- **<value>** is the value to be imported into the custom column

The **<fieldname>** attribute can be repeated for multiple custom columns.

For example, the following **<CUSTOM>** element includes import values for the CALCGROUP, MAXIMUM, and MINIMUM custom columns:

```
<CUSTOM CALCGROUP="1" MAXIMUM="10" MINIMUM="-10" />
```

---

## Interval Values

The interval values data is enclosed within the <INTS> node. Each individual interval value is enclosed within the <I> node, and consists of the V (Value) and optional S (STATUS) attributes.

**INTS:** Element containing interval values.

Elements:

**I:** Element containing a single interval value. The I node will not have an S attribute if and only if the interval status is a space.

**Attributes:**

- **V** (Interval Value): the actual recorded value for the interval
- **S** (Status Code): The status code for the interval
- **E** (Extended Status Code): The extended status code for the interval



# Appendix E

---

## Oracle Utilities Comma Separated Values (CSV) Interval Data Format

This appendix provides a detailed description of the Oracle Utilities Comma Separated Values (CSV) Interval Data Format, supported in Oracle Utilities programs such as Oracle Utilities Load Analysis, Oracle Utilities Billing Component, Oracle Utilities Rate Management, and Data Manager. This includes:

- **CSV Format**
- **Format Details**
- **Sample CSV File**

## CSV Format

This format is designed to support:

- Mixed interval lengths among cuts
- Daylight Savings Time
- International date and value formats (via tab characters for separators); the specific formats of dates and decimal values are solely dependent on the operating system in use.
- The possibility of a future enhancement for non-fixed interval lengths within cuts (with the use of Time tags for each interval).

The times that are defined in the export format will be expressed in the format designated by the **Short Date** and **Short Time** settings on the operating system. The language indicator and the decimal separator will also be read from the operating system. In all likelihood, the recipient of the export file generated by the producing Oracle Utilities application will be sharing the same language environment, so date and decimal format translation would not be necessary. If format translation is necessary, the date format, decimal separator, and language indicator found in the document description record of the export file could be used to translate the date and decimal formats to the desired format representation. The format translation that occurs in this scenario becomes solely the responsibility of the user, unless a future specification overrides the responsibility explicitly.

The format consists of two sections of data. The first section describes the export file format version and the regional settings. The second section contains data that appears in the same exact sequence as data would appear in the .LSE format, except that the data is separated by a semi-colon.

## General Description

Records in the CSV format are defined as a series of tab/comma/delimiter separated values followed by a carriage return/line feed pair. Each record begins with a record tag field that identifies that record type.

### General Rules

#### Order

Within each section, proper ordering is mandatory. Records in the Header and Detail sections should be arranged in the order as described in **Format Details** on page E-13. Trailing empty or null fields in any record cannot be omitted.

#### Fields

All fields, regardless of whether they are required or optional, must be enclosed with double quotation marks. Use the empty string to represent empty fields (“”).

#### Import Capability

The CSV Format file (\*.CSV) differs from an Enhanced Oracle Utilities (\*.LSE) file in two main respects.

- First, it contains a document description record that provides a date format and a time format. This record also contains a decimal separator, and the operating system 3-character locale identifier. Data can be re-imported from this file by translating the export date/time and decimal layouts to the appropriate format.
- Second, the data is formatted according to the date/time and decimal format that is specified in the document description record (see below).

## Document Description Record

The document description record contains the CSV format version, and the regional setting information. This required single record must precede all others in the export file. All fields within this record are required. The version number used should be related in some fashion to the version of the specification that defines the export format (e.g. 2.2). This version number must be changed when a syntactical change in the export format occurs for any reason. The importable option can be used to signify that a standard Application Program Interface (API) can process the data in this file.

```
[00000000][Document Type Identifier][Field Delimiter][Export Version][Field
Delimiter][Decimal Point Character][Field Delimiter][Language Indicator][Field
Delimiter][Date/Time Format][Field Delimiter][CSV Version][Record Delimiter]
```

### Notes:

1. The [Field Delimiter] is defined to be the character immediately following the first sequence that consists of a double quotation mark, followed by eight consecutive zero characters and then another double quotation mark ("00000000"). Characters preceding the first instance of this sequence are considered to be superfluous, and should be ignored. The [Field Delimiter] is most often a comma, but can be any delimiter character, or even non-printing characters such as the Tab character.
2. The [Document Type Identifier] identifies the format of the file.
3. The [Export Version] references the version of this document that the export was generated from. No assumptions about forward and/or backward compatibility can be assumed when comparing export files generated at differing versions. Likewise, CSV files with identical versions are said to be structurally compatible.
4. The [Decimal Point Character] cannot be the same character as the [Field Delimiter]. Should a conflict occur in the generation of the CSV file, the Decimal Separator takes precedence over the [Field Delimiter]. So, in case of a conflict, the [Field Delimiter] would be forced to change to an alternate character. Non-printable characters such as the tab character can be used in the place of commas, but please be aware that delimiters that differ from commas and semicolons may not be imported to some spreadsheets easily.
5. The [Language Indicator] is the unique 3-character abbreviation for a locale as defined by the operating system.
6. The [Date/Time Format] is expressed as a string consisting of a sequence of date and time elements strung together to form a date/time format (for example, mm/dd/yyyy hh:mm:ss). Predefined date/time format strings such as "Short Date" and "Short Time" are not legal formatting strings for the export format, because these formats are always relative to the operating system settings of the machine that generated the export, and therefore could not be determined from another computer.
7. The carriage return/line feed pair is always the [Record Delimiter]. Neither the [Field Delimiter], nor the [Decimal Separator] can be either a carriage return or a line-feed character.

## Field Relationships and Requirements

Relationships between fields exist in the CSV format. These are described along with detailed field requirements in this section. Defaults are inserted when the processing program encounters an omitted field (indicated by double commas) or white space between commas.

### Sort Code

#### Relationships:

This field has no relationship with any other field.

**Requirements:**

- Sort codes must contain eight digits, including leading 0s. Sort codes are restricted to integers between 00000000 and 99999999. However, 00000005 through 09999999 are not used, and may be ignored (see below).

**Note:** Sort code 00000000 can used **only** in the Document Description record.

- Leading white space before the sort codes is not allowed.
- The first three header records are required. The fourth header record is optional.
- The document description record starts with sort code 00000000.
- Header records will have sort code 00000001 through 00000004.
- Column 1 of each header record must contain a 0.
- Records with Sort codes 00000005 through 05000000 are reserved. Records with Sort codes 05000001 through 09999999 may be present, but will be ignored.
- Data records must start with sort code 10000000, and progress sequentially (i.e. 10000001, 10000002...) through 99999999, or as high as needed to accommodate the data, with the maximum of 99999999.
- Column 1 of the data records must be greater than 0.

**Recorder/Customer Identifier (key element)****Relationships:**

None.

**Requirements:**

- Only letters, numbers, underscores, and hyphens will be accepted in this field; all other characters and embedded blanks will cause the record to be rejected.

**Note:** Lowercase letters will be translated to uppercase before they are stored in the database.

- Must be at least one character, and no more than 64 characters in length.

**Default:**

None; field is required.

**Channel (key element)****Relationships:**

None.

**Requirements:**

- Field values must be non-negative integer only.
- Maximum value is 32767.

**Default:**

None; field is required.

**Start Time (key element)****Relationships:**

- Must be a valid date/time value less than the value of the Stop Time.

**Requirements:**

- The date format has no limitations: it is defined in the **Document Description Record Format** on page E-13.
- The date itself must correspond to acceptable date ranges of the product in use (Data Manager, Oracle Utilities Billing Component, and other non-Oracle Utilities Load Analysis applications, as well as Oracle Utilities Load Analysis).
- No spaces, colons, or any ASCII value allowed between time components.

**Default:**

None; field is required.

**Stop Time****Relationships:**

- Must be a valid date/time value greater than the value of the Start Time.
- Value must agree with the number of interval values past the start time \* SPL.
- One second is subtracted if the value is equal to an even interval start time.

**Requirements:**

See Start Time.

**Default:**

None; field is required.

**DST Participant Flag****Relationships:**

None.

**Requirements:**

Valid values are Y, N, or A, indicating whether or not this record will be processed using DST adjustments.

Flag Value	Definition
Y	(DST Participant) The intervals recorded participate in DST adjustments and the intervals during DST reflect the time change.
N	(Does not participate in DST) The intervals recorded do not make any DST time changes.
A	(DST participant) The intervals have been adjusted to 24-hour days. (At the April time change, there should be a value at the time change (a place holder) of 0 with status code "9". At the October time change, there should be a combined value.) These cuts will be converted to a "Y" or "N", and are not stored in the database as "A".

**Default:**

As specified in the INTDDEFAULTDST configuration parameter setting, the INTDCONFIG.XML file, or the CSDST value in Oracle Utilities Load Analysis.

## Invalid Record Flag

**Relationships:**

None.

**Requirements:**

Valid values are Y or N, indicating whether or not the input programs should validate the incoming record.

Flag Value	Definitions
Y	The incoming record contains unvalidated data. In Oracle Utilities Load Analysis, a value of Y can be used to run this record through the validation routines.
N	The incoming record contains valid data (mark as valid).

**Default:**

Y.

## Meter Start Reading

**Relationships:**

Must be supplied if any of the other Meter fields (Meter Stop Reading, Meter Multiplier, or Meter Offset) are supplied.

**Requirements:**

If supplied, value must be a non-negative numeric value.

**Default:**

0.

## Meter Stop Reading

**Relationships:**

Must be supplied if any of the other Meter Fields (Meter Start Reading, Meter Multiplier, or Meter Offset) are supplied.

**Requirements:**

If supplied, value must be a non-negative numeric value.

**Default:**

0 (not supplied).

## Meter Multiplier

### Relationships:

Must be supplied if any of the other Meter fields (Meter Start Reading, Meter Stop Reading, or Meter Offset) are supplied.

### Requirements:

If supplied, value must be positive numeric. However, if no value is specified, 0 will be stored. Zero (0) is not a valid value for the meter multiplier, but its presence in the database indicates that no meter multiplier was supplied.

### Default:

None, 0 stored in the database if not supplied.

**Note:** If 0 is stored and/or observed in the database and a cut is exported to the LSE format, all meter fields for that cut will be absent. A zero stored in the database is indicative of “no meter information supplied”.

## Meter Offset

### Relationships:

Must be supplied if any of the other Meter fields (Meter Start Reading, Meter Stop Reading, or Meter Multiplier) are supplied.

### Requirements:

If supplied, value must be numeric.

### Default:

0 (not supplied).

## Pulse Multiplier

### Relationships:

None.

### Requirements:

Value must be positive numeric. Also, must be supplied if Pulse Offset (see below) is supplied.

### Default:

None, 0 stored in the database if not supplied.

**Note:** If 0 is stored and/or observed in the database and a cut is exported to the LSE format, all pulse fields for that cut will be absent. A zero stored in the database is indicative of “no pulse information supplied”.

## Pulse Offset

### Relationships:

If supplied, you must also supply Pulse Multiplier.

### Requirements:

Value must be numeric.

### Default:

None.

---

## Seconds Per Interval (SPI)

**Description:**

This field will store the interval duration if appropriate. An Intervals Per Hour (IPH) of 1 can be translated to a Seconds Per Interval of 3600.

**Relationships:**

Value of 0 indicates non-uniform recording duration and a full array of interval start values (one per recording) located in the data section. (This feature will be supported in a future release and is **not currently available**.)

**Requirements:**

- Value must be a positive integer.
- Valid values for SPI are 86400, 3600, 1800, 900, 300, and 60.

**Default:**

None; field is required.

## Unit of Measure

The Oracle Utilities products will accept Units of Measure (UOMs) that are one to three digits in length. The **Units of Measure** on page B-16 are the ones supplied by Oracle.

**Description:**

See **Units of Measure** on page B-16.

**Relationships:**

None.

**Requirements:**

- Value should be a valid Unit of Measure, or a user-defined one. If the value entered is unknown to, 16 (Miscellaneous) will be stored.
- Value must be non-negative numeric.
- Maximum value is 32767.

**Default:**

None; field is required.

## Basic Unit Code

**Description:**

This field is reserved for future use.

**Relationships:**

None.

**Requirements:**

Omit this field.

**Default:**

0.

## Times Zones West of GMT

**Description:**

This value tells programs accessing this record which time zone the values have been recorded in. Its value will be the difference in time, heading west, between Greenwich Mean Time and the represented time zone as measured in half hours. Eastern Standard Time should have a value of 10; Pacific = 16. A Value of -1 will indicate "Time Zone not available."

**Relationships:**

None.

**Requirements:**

Value must be -1 or a non-negative numeric between 0 and 47, that represents the time zones West of GMT measured in half-hour increments. Any record containing a time zone greater than 47 will be rejected.

**Default:**

-1.

## Population

**Description:**

For Oracle Utilities Load Analysis Statistical records only.

**Relationships:**

None.

**Requirements:**

Value may be any non-negative integer value.

**Default:**

0.

## Weight

**Description:**

For Oracle Utilities Load Analysis Statistical records only.

**Relationships:**

None.

**Requirements:**

Value may be any non-negative numeric value.

**Default:**

0.0.

## Time Zone Standard Name

**Description:**

Defines the Time Zone to which the cut is associated. The value must be one of EST, CST, MST, PST, or be defined in the LSCALENDAR.XML configuration file. Each value in this field must contain printable ASCII characters no longer than 32 bytes.

Only letters, numbers, underscores, and hyphens will be accepted in this field, all other characters and embedded blanks will cause the record to be rejected.

---

**Note:** Lowercase letters will be translated to uppercase before being stored

**Relationships:**

Must map exactly to one of EST, CST, MST, PST or an entry in the LSCALENDAR.XML file. Oracle Utilities Load Analysis will import without checking this file.

**Requirements**

Length: <= 32 bytes

**Descriptor****Description:**

The user may enter any descriptive information in this field. Any information entered into this field will be stored in the database verbatim as supplied.

**Relationships:**

None.

**Requirements:**

- Value must be 80 characters or fewer, and may contain commas.
- May start with blanks.
- If more than 80 characters are entered, truncation will occur and the first 80 will be used.

**Timestamp****Relationships:**

None.

**Requirements:**

- Format should be YYYYMMDDHHMMSSmmm. The milliseconds (mmm) may be omitted, in which case mmm will be set to 000.
- No spaces, colons, or any ASCII value allowed between time components.

**Default:**

Current time that data is loaded into database.

**Note:** It is strongly recommended that you leave this field empty and allow the system to input the TimeStamp for you.

**Origin****Relationships:**

None.

**Requirements:**

Must be one of: M (metered), P (profiled), C (computed), or S (Oracle Utilities Load Analysis Statistic).

**Default:**

M (metered).

## Interval Value

**Description:**

This field contains the actual recorded value for a time period.

**Relationships:**

Must have one status code entry per interval data value.

**Requirements:**

Any numeric value. (Up to 15 significant digits can be stored.)

**Default:**

0 (see Status Code, Relationships, below).

## Status Code

**Description:**

This field contains the status code for the preceding Interval Value.

**Relationships:**

Must have an entry for the Interval Value.

**Requirements:**

Any character will be accepted.

**Default:**

- This field defaults to a blank ( ' ') when omitted if the interval value is present.
- This field defaults to '9' when the Interval Value field is omitted.

## Interval Start

**Description:**

- This field indicates the starting time of the recording or the time when the recording took place.
- Format must be YYYYMMDDHHMMSS. (This feature will be supported in a future release and is **not currently available**.)

**Relationships:**

- Until this feature is supported, the SPI may not be omitted (represented by „).
- There must be one entry per Interval Value.

**Requirements:**

See Start Time, Requirements.

**Default:**

None.

## Format Details

### Document Description Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	Must be 00000000 (Note: Character that immediately follows is the Field Delimiter.)
2	Document Type ID	1	LODESTAR Interval Data CSV format
3	Version	N/A	2.2
4	Decimal Separator	1	Usually "." or ","
5	Locale Abbreviation	5	Defined by OS
6	Date/Time Format	N/A	Date/Time format string

### First Header Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	Must be 00000001
2	Recorder/ Customer Identifier	64	Letters, numbers, hyphens or underscores are acceptable values
3	Channel	5	Max is 32767
4	Start Time	N/A	Cut start date and time
5	Stop Time	N/A	Cut stop date and time
6	DST Participant Flag	1	Y/N/A
7	Invalid Record Flag	1	Y/N

## Second Header Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	Must be 00000002 Default: None; field required
2	Meter Start Reading	N/A	Non-Negative Numeric Max: 9999999999999999.9999 Default: 0
3	Meter Stop Reading	N/A	Non-Negative Numeric Max: 9999999999999999.9999 Default: 0
4	Meter Multiplier	N/A	Positive Numeric (Optional) Max: 9999999999999999.9999 Default: 0
5	Meter Offset	N/A	Numeric (Optional) Max: 9999999999999999.9999 Default: 0
6	Pulse Multiplier	N/A	Positive Numeric (Optional) Max: 9999999999999999.9999 Default: 0
7	Pulse Offset	N/A	Numeric (Optional) Max: 9999999999999999.9999 Default: None; field required
8	Seconds Per Interval (SPI)	N/A	Positive Numeric Default: None; field required
9	Unit of Measure	N/A	Numeric Max: 32767 Default: None; field required
10	Basic Unit Code	N/A	Positive Numeric (Optional) Max: 9999 Default: 0
11	Time Zones West of GMT	N/A	Numeric. (Optional) Default: -1 Min: -1 Max: 47
12	Population	N/A	Positive Numeric (Optional) Max: 9999999999999999.9999 Default: 0.0
13	Weight	N/A	Positive Numeric (Optional) Max: 9999999999999999.9999 Default: 0.0
14	Time Zone Standard Name		Character (32) (Optional) If present, must be one of "EST", "CST", "MST", "PST", or definition must exist in LSCalendar.xml file.

### Third Header Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	Must be 00000003
2	Descriptor	80	

### Fourth Header Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	Must be 00000004
2	Timestamp	17	YYYYMMDDHHMMSSMM (optional) Default: current date
3	Origin	1	C, M, P, or S (optional)

### Data Record Format

ELEMENT	DESCRIPTION	MAXIMUM LENGTH IN BYTES (if applicable)	COMMENT
1	Sort Code	8	10000000 through 99999999
2	Interval Value		Interval value in locale-format
3	Status Code	1	
4	Interval Start	N/A	See Note 2
5	Interval Stop	N/A	See Note 2

### Notes

1. A data line may contain as many full data-status-time groups as desired, providing the record does not exceed a character count (record length) of 1024.
2. When this file is generated, the interval start time is always required. The interval stop time is required only for partial intervals.
3. Data spanning time must be contiguous and “missings” (or gaps) must be represented by value 0 and status '9'. For fixed-length SPIs, all intervals must be of the same duration, including “missings”. Only single intervals can appear in a single record.

## Sample CSV File

A sample \*.CSV file appears below:

```
"00000000","LODESTAR Interval Data CSV Format","2.2",".", "USE", "mm/dd/yyyy
hh:mm:ss"
"00000001","N1732","17","01/01/2000 00:00:00","01/31/2000 23:59:59","Y","Y"
"00000002","12.34","1234.56","1.0","100","1.0","0","900","04","","-1","",""
"00000003","","ACCEPTANCE TEST DATA NUMBER 1"
"00000004","","M"
"10000000","31","Q","01/01/2000 00:00:00",""
"10000001","32","Q","01/01/2000 00:15:00",""
"10000002","33","Q","01/01/2000 00:30:00",""
"10000003","34","Q","01/01/2000 00:45:00",""
"10000004","32","Q","01/01/2000 01:00:00",""
"10000005","30","Q","01/01/2000 01:15:00",""
```

## Differences with the .LSE format

There are a number of significant differences between the \*.LSE format and the \*.CSV format. These differences include:

- **Field Delimiters**
- **Field Types**
- **Document Description Record**
- **Date, Time, and Decimal Value Representation in Records**
- **Interval Data Records**

### Field Delimiters

Field delimiter in the CSV format is defined to be the character that immediately follows the first sort code ("00000000") in a file. They can be either printable or non-printable. The LSE format always uses a comma (",").

### Field Types

All fields in the CSV format must be enclosed in double quotation marks, regardless of the apparent data type. The LSE format doesn't require this.

### Document Description Record

The first record in a CSV file is always the document description record which always has a sort code of "00000000". This record provides date/time and decimal formats and locale information, as well as a version number so incompatibility can be determined. The LSE format doesn't include this record.

### Date, Time, and Decimal Value Representation in Records

All date, time, and decimal data in the CSV format uses the date/time and decimal format as defined in the document description record. The LSE file uses a specific decimal and time format.

### Interval Data Records

The interval start time is required for CSV files, but not for the \*.LSE format. Also, only a single interval value can be expressed in a record in CSV files. The interval data record contains an optional interval stop field that can be used to designate a partial interval.

## Compatibility Issues

All export files, including CSV files, have export format versions. Export files with different version numbers are inherently incompatible. A mapping algorithm is required to convert export files from one version to another. Any APIs created to read these CSV files must consult the export format version as basis for reading the files correctly. APIs should be flexible enough to react accordingly to older versions and use the proper mapping mechanism. A parser built to read a specific CSV format does not imply that it can automatically read an earlier version.



---

# Index

## Numerics

1 2-51

## A

Account class

Commercial 3-4

Residential 3-4

Account Component permissions 7-6, 7-7

Account Status records 3-2

Aggregating intervals

Aggregation methods per UOM C-16

attributes

transaction identifier 5-22

Average aggregate method 3-3, 3-5

## B

Bill Determinant Lookup codes 3-2

Bill History records 3-2

Bill History Value records 3-2

Bill History Value Table 3-2

BILLHISTORY Column 3-2

## C

CIS Account IDs 3-9

CIS Account Table 3-9

CISFORMAT.TXT 9-3

Code

in Jurisdiction Lookup code record 3-3

Control File 9-2, 9-3

Currency Presentation 1-6

Customer Component permissions 7-6

## D

Data Sources tab and functions

enabling 7-2

Database schema diagram A-1

Database sort order D-2

Date Presentation 1-6

Deletion of data sources

enabling 7-2

Deletion of positions

enabling 7-2

Deletion of roles

enabling 7-2

Deletion of sessions

enabling 7-2

Deletion of users

enabling 7-2

Directory Code field 3-10

DST adjustments D-5

## E

EMail field 3-10

End Use Lookup codes 3-3

Enhanced Direct Input File C-15

## F

Factor Table 3-11

Factor Value record 3-11

Factor value records 3-11

Fax field 3-10

Forecast Management Overview 1-1

## G

Group Category Lookup codes 3-12

## H

How to

Modify the .tls file 2-51

## I

ID field 3-9

Identifier 3-2

Insertion of accounts

enabling 7-6

INTDIMP

Inputs 8-13

International

Database Support 1-9

File Formats 1-9

International Support 1-5

Interrogations Lookup codes 3-12

Interval Data Format D-1

## J

Jurisdiction Code field 3-9, 3-11

## K

- kvarh 3-5
- KWH 3-5
- KWH-IN 2-21, 3-5
- KWH-OUT 2-21, 3-5

## L

- LAGGING KVARH 3-5
- Languages 1-5
- LEADING KVARH 3-5
- List Management menu item and function
  - enabling 7-8
- List Management Permission 7-8
- Locales 1-5
- Locales function
  - enabling 7-4
- Locales Permission 7-4
- LODESTAR.CFG 9-2
- Lookups
  - Account Status 3-2
  - Bill Determinant 3-2
  - Jurisdiction 3-2
  - LSCurrency 3-2
  - Operating Company 3-2
  - Revenue Code 3-2
  - SIC 3-2
  - UOM (Unit of Measure) 3-2
- LSCurrency table 1-6
- LTMH Monitor 6-77
- LTMH Server 6-76

## M

- Max aggregate method 3-3, 3-5
- Messaging Tool Tab 6-78
- Meter nodes D-6
- Meter Value records 3-2
- Multiple Currency Support 1-5, 1-8
  - Oracle Utilities Receivables Component 1-8
  - Oracle Utilities Receivables Component Reports 1-8
  - Other Oracle Utilities Products 1-8

## N

- Name
  - in Jurisdiction Lookup code record 3-3
- Name of File field 3-10
- Node Descriptions D-2
- Null value 3-9
- Numeric Presentation 1-6

## O

- Operating Company
  - definition 3-4
- Operating Company Code field 3-9, 3-11
- Oracle Utilities UOM codes 3-5
- Output File 9-2
- Override Code field 3-15
- Override Lookup codes 3-15

## P

- Pager field 3-10
- Permissions
  - Security 7-2
- Phone #1 field 3-10
- Phone #2 field 3-10
- Postwindow field 3-12
- Preview field 3-12
- Print Detail 9-9
- Prorate charges? 3-11

## R

- Relationships between nodes D-4
- Report Admin menu option
  - enabling 7-9
- Revenue Code Lookup codes 3-4
- RUNRS
  - Inputs 8-41
- Runtime Queue Tab 6-78
- Runtime Services Tab 6-77

## S

- Search Account menu item and function
  - enabling 7-5
- Search Customer menu item and function
  - enabling 7-5
- Search Meters menu item and function
  - enabling 7-6
- Search Recorders menu item and function
  - enabling 7-5
- Sessions tab and functions
  - enabling 7-2
- Standard Direct Input File C-15
- Standard Industrial Classification (SIC) Lookup codes 3-5
- Starting the LTMH Server 6-76
- Stop Billing
  - Automatic Only 5-13, 5-20
  - Manual Only 5-13, 5-21
- Stope Billing
  - for both Automatic and Approval Required modes 5-13, 5-21

## T

- tail attributes 5-22
- Title field 3-10
- Total aggregate method 3-3, 3-5
- Transaction File 9-2, 9-8
- Transaction File format
  - creating 9-1

## U

- Unit of Measure 3-2
- Unit of Measure (UOM) Lookup code 3-5
- Unit of Measure field 3-11
- UOM 3-2, 3-11
- User Context function
  - enabling 7-4
- User Context Permission 7-4
- Users tab and functions
  - enabling 7-2

# X

XML Interval Data Format D-1

