

**Oracle® Enterprise Manager**

Extensibility

10g Release 2 (10.2) for Windows or UNIX

**B16246-01**

February 2006

Oracle Enterprise Manager Extensibility, 10g Release 2 (10.2) for Windows or UNIX

B16246-01

Copyright © 2003, 2006, Oracle. All rights reserved.

The Programs (which include both the software and documentation) contain proprietary information; they are provided under a license agreement containing restrictions on use and disclosure and are also protected by copyright, patent, and other intellectual and industrial property laws. Reverse engineering, disassembly, or decompilation of the Programs, except to the extent required to obtain interoperability with other independently created software or as specified by law, is prohibited.

The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

If the Programs are delivered to the United States Government or anyone licensing or using the Programs on behalf of the United States 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, use, duplication, disclosure, modification, and adaptation of the Programs, including documentation and technical data, shall be subject to the licensing restrictions set forth in the applicable Oracle license agreement, and, to the extent applicable, the additional rights set forth in FAR 52.227-19, Commercial Computer Software—Restricted Rights (June 1987). Oracle Corporation, 500 Oracle Parkway, Redwood City, CA 94065

The Programs are not intended for use in any nuclear, aviation, mass transit, medical, or other inherently dangerous applications. It shall be the licensee's responsibility to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of such applications if the Programs are used for such purposes, and we disclaim liability for any damages caused by such use of the Programs.

Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

The Programs may provide links to Web sites and access to content, products, and services from third parties. Oracle is not responsible for the availability of, or any content provided on, third-party Web sites. You bear all risks associated with the use of such content. If you choose to purchase any products or services from a third party, the relationship is directly between you and the third party. Oracle is not responsible for: (a) the quality of third-party products or services; or (b) fulfilling any of the terms of the agreement with the third party, including delivery of products or services and warranty obligations related to purchased products or services. Oracle is not responsible for any loss or damage of any sort that you may incur from dealing with any third party.

---

---

# Contents

<b>Preface</b> .....	xi
Audience .....	xi
Documentation Accessibility .....	xi
Related Documents .....	xii
Conventions .....	xii

## Part I Extensibility

### 1 Extending Monitoring

1.1	What You Get .....	1-1
1.2	What is a Management Plug-in.....	1-2
1.3	Management Plug-ins Available from the Oracle Technology Network.....	1-3
1.4	Management Plug-in Lifecycle .....	1-3
1.5	About the Extensibility Guide.....	1-5

### 2 Developing a Management Plug-in

2.1	Designing Your Management Plug-in .....	2-2
2.2	Developing Requisite Management Plug-in Files.....	2-2
2.2.1	Target Definition Files.....	2-3
2.2.1.1	Creating the target type metadata file.....	2-3
2.2.1.2	Validate your new target type definitions.....	2-10
2.2.1.3	Creating a Default Collection File .....	2-12
2.2.1.4	Adding SYSTEM Reports (Optional).....	2-14
2.2.1.5	Adding Related Links to Target Home Pages.....	2-14
2.2.1.6	Additional Scripts and Binaries.....	2-15
2.2.2	Ensuring Accurate XML .....	2-15
2.2.3	Development Guidelines .....	2-16
2.2.3.1	Guidelines for Defining Target Metadata .....	2-16
2.2.3.2	Guidelines for Defining Collections .....	2-18
2.3	Creating a Management Plug-In Archive .....	2-19
2.4	Uploading the Management Plug-in Archive into Enterprise Manager .....	2-20
2.5	Adding a Target Instance.....	2-22
2.6	Viewing Results.....	2-22
2.7	Troubleshooting Management Plug-ins .....	2-23

### 3 Validating XML

3.1	What is ILINT? .....	3-1
3.1.1	What types of validation does ILINT perform? .....	3-1
3.1.1.1	Static XML Validation .....	3-1
3.1.1.2	Dynamic Validation .....	3-2
3.1.1.3	Metadata Version Checking.....	3-2
3.1.2	ILINT Output .....	3-3
3.2	Before Using ILINT.....	3-3
3.3	Using ILINT .....	3-4
3.3.1	ILINT Examples .....	3-5
3.3.1.1	Static Validation.....	3-5
3.3.1.2	Dynamic Validation .....	3-6
3.3.1.3	Checking Metadata Version Compatibility .....	3-6
3.3.1.4	Generating Syntactically Correct XML.....	3-7
3.3.2	Usage Notes.....	3-7

### 4 Adding Reports

4.1	What You Get .....	4-1
4.1.1	Reports Page (Target Home Page) .....	4-1
4.1.2	Report Definitions Page .....	4-2
4.2	Report Definition File .....	4-3
4.3	Creating a Report Definition File.....	4-4
4.3.1	Report Definition File Development Process .....	4-4
4.3.2	Report Lifecycle: Updating Report Definitions.....	4-9
4.4	PL/SQL Application Programmer Interface .....	4-9
4.4.1	PL/SQL Methods for Creating Report Definitions .....	4-9
4.4.1.1	mgmt_ip.create_report_definition .....	4-9
4.4.1.2	mgmt_ip.add_element_to_report_def.....	4-10
4.4.1.3	mgmt_mp_homepage.add_report .....	4-11
4.4.1.4	mgmt_view_util.adjust_tz.....	4-12
4.4.2	PL/SQL Type Definitions.....	4-12
4.4.3	Element Parameters.....	4-13
4.4.3.1	Table Element Parameters.....	4-13
4.4.3.2	Chart Element .....	4-19
4.4.4	Metric Details Element.....	4-22
4.4.5	Text Element Parameters.....	4-24
4.4.6	Report-Wide Parameters .....	4-24
4.5	Development Guidelines .....	4-25

### 5 Management Using Web Services

5.1	Monitoring Using Web Services in Enterprise Manager.....	5-1
5.2	Generating Target Metadata and Collection Files .....	5-2
5.2.1	Web Services Command Line Tool .....	5-2
5.2.1.1	Security.....	5-2
5.2.1.2	Syntax .....	5-3
5.2.1.3	Usage .....	5-3

5.3	Creating a Management Plug-in Archive.....	5-7
5.4	Importing the Management Plug-in .....	5-8
5.5	Deploying the Management Plug-in to the Management Agent.....	5-9
5.6	Adding a Target Instance.....	5-10
5.7	Viewing Monitored Metrics .....	5-10

## Part II Reference

### 6 Management Repository Views

6.1	Overview .....	6-1
6.2	Monitoring Views .....	6-3
6.2.1	MGMT\$METRIC_ERROR_HISTORY .....	6-3
6.2.2	MGMT\$METRIC_ERROR_CURRENT.....	6-3
6.2.3	MGMT\$TARGET_COMPONENTS.....	6-4
6.2.4	MGMT\$BLACKOUT_HISTORY .....	6-5
6.2.5	MGMT\$BLACKOUTS.....	6-6
6.2.6	MGMT\$ALERT_ANNOTATIONS .....	6-7
6.2.7	MGMT\$ALERT_NOTIF_LOG.....	6-8
6.2.8	MGMT\$TARGET_METRIC_COLLECTIONS.....	6-10
6.2.9	MGMT\$TARGET_METRIC_SETTINGS .....	6-11
6.2.10	MGMT\$AVAILABILITY_CURRENT .....	6-14
6.2.11	MGMT\$AVAILABILITY_HISTORY .....	6-14
6.2.12	MGMT\$ALERT_CURRENT.....	6-15
6.2.13	MGMT\$ALERT_HISTORY .....	6-17
6.2.14	MGMT\$METRIC_DETAILS.....	6-19
6.2.15	MGMT\$METRIC_CURRENT .....	6-20
6.2.16	MGMT\$METRIC_HOURLY .....	6-22
6.2.17	MGMT\$METRIC_DAILY .....	6-23
6.3	Inventory Views .....	6-24
6.3.1	MGMT\$TARGET .....	6-24
6.3.2	MGMT\$TARGET_TYPE .....	6-26
6.3.3	MGMT\$TARGET_TYPE_DEF .....	6-27
6.3.4	MGMT\$TARGET_ASSOCIATIONS.....	6-27
6.3.5	MGMT\$TARGET_MEMBERS .....	6-28
6.3.6	MGMT\$TARGET_FLAT_MEMBERS .....	6-28
6.3.7	MGMT\$TARGET_TYPE_PROPERTIES .....	6-29
6.3.8	MGMT\$TARGET_PROPERTIES.....	6-29
6.3.9	MGMT\$METRIC_CATEGORIES .....	6-30
6.4	Policy Definition Views.....	6-30
6.4.1	MGMT\$POLICIES .....	6-30
6.4.2	MGMT\$POLICY_PARAMETERS .....	6-32
6.4.3	MGMT\$POLICY_VIOLATION_CTXT.....	6-32
6.4.4	MGMT\$TARGET_POLICY_EVAL_SUMM .....	6-33
6.4.5	MGMT\$POLICY_VIOL_ANNOTATIONS.....	6-33
6.4.6	MGMT\$POLICY_VIOL_NOTIF_LOG .....	6-34
6.5	Policy Association Views.....	6-35

6.5.1	MGMT\$TARGET_POLICIES.....	6-35
6.5.2	MGMT\$TARGET_POLICY_SETTINGS.....	6-36
6.6	Policy Violation Views.....	6-37
6.6.1	MGMT\$POLICY_VIOLATION_CURRENT.....	6-37
6.6.2	MGMT\$POLICY_VIOLATION_HISTORY.....	6-39
6.6.3	MGMT\$POLICY_VIOLATION_CONTEXT.....	6-40
6.7	Management Template Views.....	6-40
6.7.1	MGMT\$TEMPLATES.....	6-41
6.7.2	MGMT\$TEMPLATE_POLICY_SETTINGS.....	6-41
6.7.3	MGMT\$TEMPLATE_METRICCOLLECTION.....	6-42
6.7.4	MGMT\$TEMPLATE_METRIC_SETTINGS.....	6-43
6.8	Job Views.....	6-45
6.8.1	MGMT\$JOBS.....	6-45
6.8.2	MGMT\$JOB_TARGETS.....	6-46
6.8.3	MGMT\$JOB_EXECUTION_HISTORY.....	6-47
6.8.4	MGMT\$JOB_STEP_HISTORY.....	6-47
6.8.5	MGMT\$JOB_ANNOTATIONS.....	6-48
6.8.6	MGMT\$JOB_NOTIFICATION_LOG.....	6-49
6.9	Application Service Level Management Views.....	6-50
6.9.1	MGMT\$CSM_REGION.....	6-50
6.9.2	MGMT\$CSM_WATCHLIST.....	6-50
6.9.3	MGMT\$CSM_METRIC_DETAILS.....	6-51
6.9.4	MGMT\$CSM_MT_METRIC_DETAILS.....	6-52
6.9.5	MGMT\$CSM_URL_HOURLY.....	6-52
6.9.6	MGMT\$CSM_URL_DAILY.....	6-53
6.9.7	MGMT\$CSM_URL_DIST_HOURLY.....	6-54
6.9.8	MGMT\$CSM_URL_DIST_DAILY.....	6-55
6.9.9	MGMT\$CSM_MT_URL_HOURLY.....	6-55
6.9.10	MGMT\$CSM_MT_URL_DAILY.....	6-56
6.9.11	MGMT\$CSM_MT_URL_DIST_HOURLY.....	6-57
6.9.12	MGMT\$CSM_MT_URL_DIST_DAILY.....	6-58
6.9.13	MGMT\$CSM_IP_HOURLY.....	6-59
6.9.14	MGMT\$CSM_IP_DAILY.....	6-60
6.9.15	MGMT\$CSM_IP_DIST_HOURLY.....	6-61
6.9.16	MGMT\$CSM_IP_DIST_DAILY.....	6-61
6.9.17	MGMT\$CSM_MT_IP_HOURLY.....	6-62
6.9.18	MGMT\$CSM_MT_IP_DAILY.....	6-63
6.9.19	MGMT\$CSM_MT_IP_DIST_HOURLY.....	6-64
6.9.20	MGMT\$CSM_MT_IP_DIST_DAILY.....	6-65
6.9.21	MGMT\$CSM_DOMAIN_HOURLY.....	6-65
6.9.22	MGMT\$CSM_DOMAIN_DAILY.....	6-66
6.9.23	MGMT\$CSM_DOMAIN_DIST_HOURLY.....	6-67
6.9.24	MGMT\$CSM_DOMAIN_DIST_DAILY.....	6-68
6.9.25	MGMT\$CSM_SUBNET_HOURLY.....	6-68
6.9.26	MGMT\$CSM_SUBNET_DAILY.....	6-69
6.9.27	MGMT\$CSM_SUBNET_DIST_HOURLY.....	6-70
6.9.28	MGMT\$CSM_SUBNET_DIST_DAILY.....	6-70

6.9.29	MGMT\$E2E_1DAY .....	6-71
6.9.30	MGMT\$E2E_HOURLY .....	6-71
6.9.31	MGMT\$E2E_RAW.....	6-72
6.10	Configuration Views .....	6-73
6.10.1	MGMT\$DB_TABLESPACES.....	6-73
6.10.2	MGMT\$DB_DATAFILES .....	6-74
6.10.3	MGMT\$DB_CONTROLFILES .....	6-74
6.10.4	MGMT\$DB_DBNINSTANCEINFO.....	6-75
6.10.5	MGMT\$DB_FEATUREUSAGE .....	6-76
6.10.6	MGMT\$DB_INIT_PARAMS .....	6-77
6.10.7	MGMT\$DB_LICENSE.....	6-77
6.10.8	MGMT\$DB_REDOLOGS.....	6-78
6.10.9	MGMT\$DB_ROLLBACK_SEGS.....	6-79
6.10.10	MGMT\$DB_SGA .....	6-80
6.10.11	MGMT\$DB_TABLESPACES.....	6-80
6.10.12	MGMT\$DB_OPTIONS.....	6-81
6.11	Oracle Home Patching Views .....	6-81
6.11.1	MGMT\$EM_HOMES_PLATFORM.....	6-82
6.11.2	MGMT\$HOMES_AFFECTED.....	6-82
6.11.3	MGMT\$APPL_PATCH_AND_PATCHSET .....	6-82
6.11.4	MGMT\$APPLIED_PATCHES .....	6-82
6.11.5	MGMT\$APPLIED_PATCHSETS.....	6-83
6.12	Linux Patching Views.....	6-83
6.12.1	MGMT\$HOSTPATCH_HOSTS.....	6-83
6.12.2	MGMT\$HOSTPATCH_GROUPS.....	6-84
6.12.3	MGMT\$HOSTPATCH_GRP_COMPL_HIST .....	6-84
6.12.4	MGMT\$HOSTPATCH_HOST_COMPL .....	6-84
6.13	Security Views .....	6-84
6.13.1	MGMT\$ESA_ALL_PRIVS_REPORT .....	6-84
6.13.2	MGMT\$ESA_ANY_DICT_REPORT .....	6-85
6.13.3	MGMT\$ESA_ANY_PRIV_REPORT .....	6-85
6.13.4	MGMT\$ESA_AUDIT_SYSTEM_REPORT .....	6-85
6.13.5	MGMT\$ESA_BECOME_USER_REPORT .....	6-86
6.13.6	MGMT\$ESA_CATALOG_REPORT.....	6-86
6.13.7	MGMT\$ESA_CONN_PRIV_REPORT.....	6-86
6.13.8	MGMT\$ESA_CREATE_PRIV_REPORT .....	6-87
6.13.9	MGMT\$ESA_DBA_GROUP_REPORT.....	6-87
6.13.10	MGMT\$ESA_DBA_ROLE_REPORT .....	6-87
6.13.11	MGMT\$ESA_DIRECT_PRIV_REPORT.....	6-88
6.13.12	MGMT\$ESA_EXMPT_ACCESS_REPORT.....	6-88
6.13.13	MGMT\$ESA_KEY_OBJECTS_REPORT .....	6-88
6.13.14	MGMT\$ESA_OH_OWNERSHIP_REPORT .....	6-89
6.13.15	MGMT\$ESA_OH_PERMISSION_REPORT.....	6-89
6.13.16	MGMT\$ESA_POWER_PRIV_REPORT.....	6-89
6.13.17	MGMT\$ESA_PUB_PRIV_REPORT .....	6-90
6.13.18	MGMT\$ESA_SYS_PUB_PKG_REPORT.....	6-90
6.13.19	MGMT\$ESA_TABSP_OWNERS_REPORT.....	6-90

6.13.20	MGMT\$ESA_TRC_AUD_PERM_REPORT .....	6-90
6.13.21	MGMT\$ESA_WITH_ADMIN_REPORT .....	6-91
6.13.22	MGMT\$ESA_WITH_GRANT_REPORT .....	6-91
6.14	Configuration Management Views .....	6-91
6.14.1	MGMT\$CSA_COLLECTIONS.....	6-91
6.14.2	MGMT\$CSA_FAILED .....	6-95
6.14.3	MGMT\$CSA_HOST_OS_COMPONENTS.....	6-95
6.14.4	MGMT\$CSA_HOST_SW .....	6-96
6.14.5	MGMT\$CSA_HOST_COOKIES .....	6-96
6.14.6	MGMT\$CSA_HOST_CUSTOM.....	6-97
6.14.7	MGMT\$CSA_HOST_RULES .....	6-97
6.14.8	MGMT\$CSA_HOST_CPUS.....	6-97
6.14.9	MGMT\$CSA_HOST_IOCARDS.....	6-98
6.14.10	MGMT\$CSA_HOST_NICS.....	6-98
6.14.11	MGMT\$CSA_HOST_OS_PROPERTIES.....	6-99
6.14.12	MGMT\$CSA_HOST_OS_FILESYSTEMS .....	6-99
6.14.13	MGMT\$ECM_CONFIG_HISTORY .....	6-99
6.14.14	MGMT\$ECM_CONFIG_HISTORY_KEY1 .....	6-100
6.14.15	MGMT\$ECM_CONFIG_HISTORY_KEY2 .....	6-101
6.14.16	MGMT\$ECM_CONFIG_HISTORY_KEY3 .....	6-101
6.14.17	MGMT\$ECM_CONFIG_HISTORY_KEY4 .....	6-102
6.14.18	MGMT\$ECM_CONFIG_HISTORY_KEY5 .....	6-102
6.14.19	MGMT\$ECM_CONFIG_HISTORY_KEY6 .....	6-103
6.14.20	MGMT\$HW_NIC .....	6-104
6.14.21	MGMT\$OS_COMPONENTS.....	6-104
6.14.22	MGMT\$OS_FS_MOUNT.....	6-104
6.14.23	MGMT\$OS_HW_SUMMARY .....	6-105
6.14.24	MGMT\$OS_KERNEL_PARAMS.....	6-105
6.14.25	MGMT\$OS_PATCHES .....	6-105
6.14.26	MGMT\$OS_SUMMARY.....	6-106
6.14.27	MGMT\$SOFTWARE_COMP_PATCHSET.....	6-106
6.14.28	MGMT\$SOFTWARE_COMPONENT_ONEOFF.....	6-106
6.14.29	MGMT\$SOFTWARE_COMPONENTS.....	6-107
6.14.30	MGMT\$SOFTWARE_DEPENDENCIES.....	6-107
6.14.31	MGMT\$SOFTWARE_HOMES .....	6-108
6.14.32	MGMT\$SOFTWARE_ONEOFF_PATCHES.....	6-108
6.14.33	MGMT\$SOFTWARE_OTHERS.....	6-108
6.14.34	MGMT\$SOFTWARE_PATCHES_IN_HOMES.....	6-109
6.14.35	MGMT\$SOFTWARE_PATCHSETS.....	6-109
6.15	Database Cluster Views .....	6-109
6.15.1	MGMT\$CLUSTER_INTERCONNECTS .....	6-110
6.15.2	MGMT\$RACDB_INTERCONNECTS .....	6-110
6.15.3	MGMT\$HA_BACKUP.....	6-111
6.16	Storage Reporting Views .....	6-112
6.16.1	MGMT\$STORAGE_REPORT_DATA.....	6-112
6.16.2	MGMT\$STORAGE_REPORT_KEYS .....	6-113
6.16.3	MGMT\$STORAGE_REPORT_PATHS.....	6-113



6.16.4	MGMT\$STORAGE_REPORT_ISSUES .....	6-114
6.16.5	MGMT\$STORAGE_REPORT_DISK .....	6-114
6.16.6	MGMT\$STORAGE_REPORT_VOLUME.....	6-115
6.16.7	MGMT\$STORAGE_REPORT_LOCALFS .....	6-115
6.16.8	MGMT\$STORAGE_REPORT_NFS.....	6-116

## 7 Fetchlets

7.1	OS Command Fetchlets.....	7-1
7.1.1	OS Fetchlet .....	7-1
7.1.2	OSLines Fetchlet (split into lines).....	7-3
7.1.3	OSLineToken Fetchlet (tokenized lines).....	7-5
7.2	SQL Fetchlet .....	7-7
7.3	SNMP Fetchlet.....	7-12
7.4	URL Timing Fetchlet .....	7-15
7.5	Dynamic Monitoring Service (DMS) Fetchlet.....	7-21
7.5.1	Advantages to Using DMS for Oracle Management Agent Integration .....	7-21
7.5.2	DMS Fetchlet/Oracle Management Agent Integration Instructions .....	7-23
7.5.2.1	Integrating DMS Data with the Management Agent .....	7-24
7.6	HTTP Data Fetchlets.....	7-26
7.6.1	URL Fetchlet (raw).....	7-26
7.6.2	URL Lines Fetchlet (split into lines).....	7-27
7.6.3	URL Line Token Fetchlet (tokenized lines).....	7-28
7.7	URLXML Fetchlet .....	7-29
7.8	WBEM Fetchlet.....	7-31
7.9	JDBC Fetchlet.....	7-35
7.10	OJMX/SOAP Fetchlet .....	7-36

## Index



---

---

# Preface

This manual covers Enterprise Manager framework extensibility and related reference information.

Note that more recent versions of this and other Enterprise Manager books may be available on the Oracle Technology Network:

<http://www.oracle.com/technology/documentation/oem.html>

## Audience

This guide is written for developers or administrators needing to extend Enterprise Manager's monitoring capability by defining custom target types. You should already be familiar with Enterprise Manager administrative tasks you want to perform.

You should also have a working knowledge of XML and DTDs, as well as being familiar with the operation of your specific UNIX or Windows system. Refer to your platform-specific documentation, if necessary.

## Documentation Accessibility

Our goal is to make Oracle products, services, and supporting documentation accessible, with good usability, to the disabled community. To that end, our documentation includes features that make information available to users of assistive technology. This documentation is available in HTML format, and contains markup to facilitate access by the disabled community. Accessibility standards will continue to evolve over time, and Oracle is actively engaged with other market-leading technology vendors to address technical obstacles so that our documentation can be accessible to all of our customers. For more information, visit the Oracle Accessibility Program Web site at

<http://www.oracle.com/accessibility/>

### Accessibility of Code Examples in Documentation

Screen readers may not always correctly read the code examples in this document. The conventions for writing code require that closing braces should appear on an otherwise empty line; however, some screen readers may not always read a line of text that consists solely of a bracket or brace.

### Accessibility of Links to External Web Sites in Documentation

This documentation may contain links to Web sites of other companies or organizations that Oracle does not own or control. Oracle neither evaluates nor makes any representations regarding the accessibility of these Web sites.

## Related Documents

For more information, see the following manuals in the Oracle Enterprise Manager 10g Release 2 documentation set:

- *Oracle Enterprise Manager Administrator's Guide*
- *Oracle Enterprise Manager Concepts*
- *Oracle Enterprise Manager Grid Control Quick Installation Guide*
- *Oracle Enterprise Manager Grid Control Installation and Basic Configuration*
- *Oracle Enterprise Manager Configuration for Oracle Collaboration Suite*
- *Oracle Enterprise Manager Policy Reference Manual*
- *Oracle Enterprise Manager Metric Reference Manual*
- *Oracle Enterprise Manager Command Line Interface*

## Conventions

The following text conventions are used in this document:

<b>Convention</b>	<b>Meaning</b>
<b>boldface</b>	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
<i>italic</i>	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
monospace	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

# Part I

---

## Extensibility

This section of the guide provides instruction on extending the Enterprise Manager framework.

Part I contains the following chapters:

- [Chapter 1, "Extending Monitoring"](#)
- [Chapter 2, "Developing a Management Plug-in"](#)
- [Chapter 3, "Validating XML"](#)
- [Chapter 4, "Adding Reports"](#)
- [Chapter 5, "Management Using Web Services"](#)



---

---

# Extending Monitoring

Out-of-box, Enterprise Manager can monitor the most common hardware and applications (target types) used in enterprise environments. Because it is not possible to anticipate all possible target types that may exist in your IT environment, Enterprise Manager provides a modular way to extend monitoring capabilities called Management Plug-ins.

A Management Plug-in lets you create custom target types that allow you to monitor applications or hardware unique to your enterprise directly from the Enterprise Manager console. You simply deploy the Management Plug-in to Management Agents throughout your enterprise.

This chapter covers the following topics:

- [What You Get](#)
- [What is a Management Plug-in](#)
- [Management Plug-ins Available from the Oracle Technology Network](#)
- [Management Plug-in Lifecycle](#)
- [About the Extensibility Guide](#)

## 1.1 What You Get

Using Management Plug-ins to define custom target types to be monitored by Enterprise Manager, you are able to centralize all of your management information in the console. By default, Enterprise Manager management and monitoring functionality is automatically extended to target instances of the type defined by your Management Plug-in. For example:

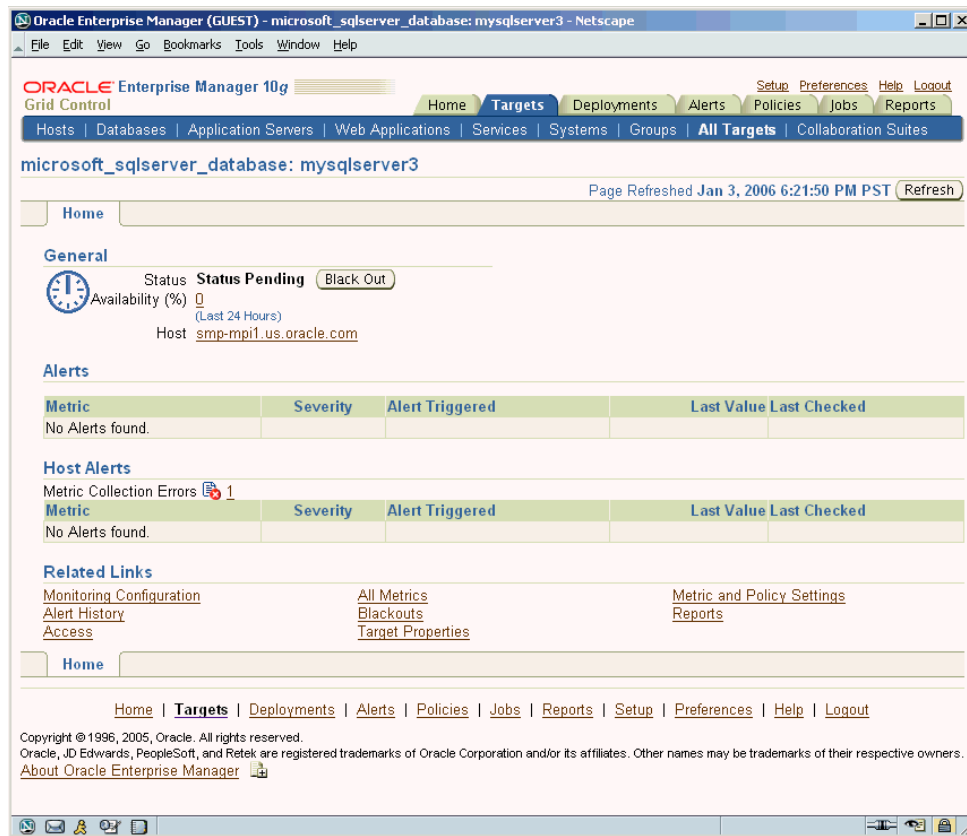
- Core framework functionality such as Alerts, Policies, Blackouts, and Management Templates, Jobs
- Groups and Systems
- Configuration Management
- SYSTEM Reports

These and other features allow you to better diagnose availability and performance problems by allowing you to correlate problems across your enterprise.

### Target Home Page

Each target instance added to Enterprise Manager is automatically provided with a comprehensive target type-specific home page. No coding is required. The default target home page provides a consolidated view of target availability and alerts. Links to related monitoring information such as Monitoring Configuration, Metric and Policy Settings, and Alert History are also included. Figure 1-1 shows the default target home page for a Microsoft SQL Server database.

Figure 1-1 Microsoft SQL Server Home Page



## 1.2 What is a Management Plug-in

A Management Plug-in is a group of target definition files that has been added to a Management Plug-in Archive using the Enterprise Manager Command Line Interface (EM CLI). Only when the target definition files have been added to an archive are they officially a Management Plug-in. A Management Plug-in Archive file (.jar file) associates the target definition files together as a true Management Plug-in and may consist of multiple Management Plug-ins.

A Management Plug-in consists of several types of files that serve specific functions at different tiers within the Enterprise Manager framework. The target type metadata file is an integral part of defining a new target type and is required by the EM CLI, in addition to a default collection file, to create a new Management Plug-in within the Management Plug-in Archive. The target metadata and collection file, at a minimum, are required to create a new Management Plug-in. There are, however, other supported file types that may be needed for target type-specific monitoring and reporting functions.



## 1.3 Management Plug-ins Available from the Oracle Technology Network

Ready-to-use Management Plug-ins are available from the Oracle Technology Network (OTN) at the Oracle Enterprise Manager 10g Grid Control Extensions Exchange located at:

<http://www.oracle.com/technology/products/oem/extensions/index.html>

This site is your central information source for all Enterprise Manager extensibility information. In addition to tutorials and the latest documentation, you can download ready-to-use Management Plug-ins developed by Oracle as well as third-party integrators.

Management Plug-ins that are currently available include:

- BEA WebLogic Plug-in
- Check Point Firewall Plug-in
- F5 BIG-IP Load Balancer Plug-in
- IBM DB2 Database Plug-in
- IBM WebSphere Plug-in
- Juniper Netscreen Firewall Plug-in
- Microsoft SQL Server Plug-in
- NetApp Filer Plug-in

Because this list is continually being updated, you should check the Extensions Exchange site regularly.

## 1.4 Management Plug-in Lifecycle

As with any custom code implementation, a Management Plug-in has two distinct lifecycles: Development and Deployment.

### Development

The development lifecycle, which is the primary focus of this guide, consists of four stages:

1. **Design:** Determine what metrics are required to monitor your target, threshold values for alerts, and collection schedules.
2. **Develop:** Create the methods used to retrieve the metrics by defining the target type metadata and collection files, and any required support files such as monitoring scripts and/or report definition files.
3. **Validate:** Using the supplied XML validation tool (ILINT), validate the XML used to define the target metadata and collection files. Using the Management Agent's Metric Browser, you verify metric data.
4. **Package:** Create a Management Plug-in Archive using the EM CLI.

### Deployment

The deployment lifecycle is managed from the Enterprise Manager console and consists of the following four stages:

1. Import the Management Plug-in from the Management Plug-in Archive.
2. Deploy the Management Plug-in to one or more Management Agents.

3. Add or delete target instances defined by your Management Plug-in from the Management Agent home page(s).
4. Upgrade/Downgrade Management Plug-ins. The Management Agent uses the latest Management Plug-in version that has been deployed.

---

---

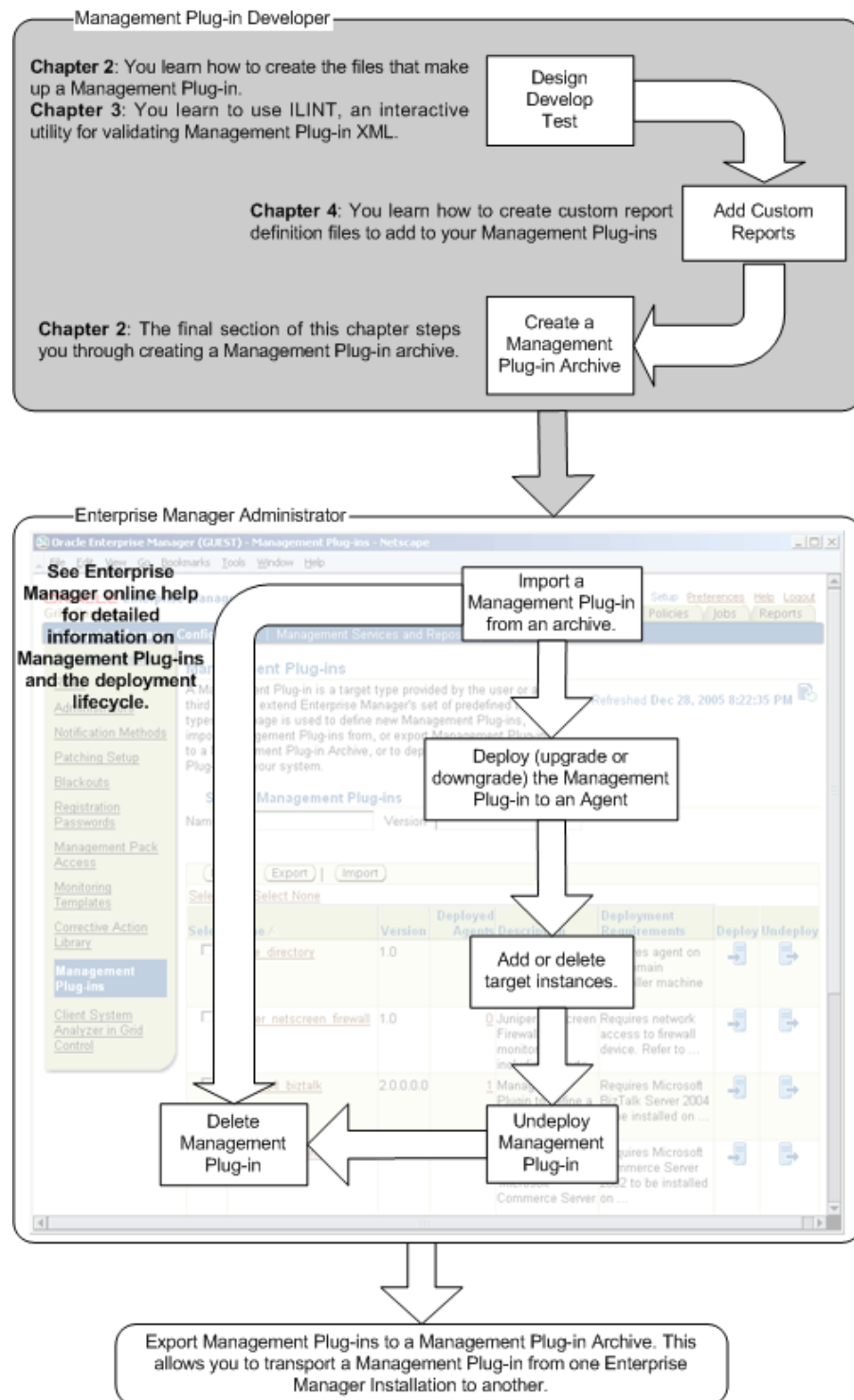
**Note:** See Enterprise Manager online help for information on Management Plug-ins and the deployment lifecycle.

---

---

[Figure 1-2](#) shows the complete Management Plug-in lifecycle.

Figure 1–2 Management Plug-in Lifecycle: Combined Development and Deployment



## 1.5 About the Extensibility Guide

The Extensibility guide assumes you have sufficient familiarity with the Enterprise Manager Grid Control framework, its components, and functionality. In addition, familiarity with XML and PL/SQL is assumed. For more information about Enterprise Manager Grid Control, see the Enterprise Manager Concepts guide and the Enterprise

Manager Grid Control online help. Additional information is available from the Enterprise Manager documentation area on OTN as well as the Extensions Exchange.

**[HTTP://www.oracle.com/technology/products/oem/extensions/index.html](http://www.oracle.com/technology/products/oem/extensions/index.html)**

The Extensibility guide is divided into two sections. The first section provides detailed instructions for creating your Management Plug-in and the second provides reference information that may be required when developing your Management Plug-in. The remaining chapters of this guide consist of the following:

- **Chapter 2:** Instructions on how to define the required target metadata and collection files. Using the Management Agent's Metric Browser is also covered in this chapter.
- **Chapter 3:** Using ILINT to validate the XML used to create your target metadata and collection files.
- **Chapter 4:** Instructions on how to create a report definition file for inclusion in your Management Plug-in. The complete Information Publisher PL/SQL API is documented in this chapter.
- **Chapter 5:** Instructions on using the OJMX fetchlet to monitor targets exposing a Web Service management interface.
- **Chapter 6:** Reference for all management views used to access information stored in Enterprise Manager's Management Repository.
- **Chapter 7:** Reference for available fetchlets (metric data extraction mechanisms).

---

---

## Developing a Management Plug-in

Each Management Plug-in defines a new type of target that can be monitored by Enterprise Manager. A target, or more specifically, a target instance, can be defined as any monitorable entity within an enterprise. A monitorable entity can be an application running on a server, the server itself, the network or any of its constituent parts. Enterprise Manager makes managing target instances simple by allowing you to add new target instances to the management framework from the Enterprise Manager Grid Control console, at which point you can take advantage of Enterprise Manager's monitoring and administrative functionality.

Enterprise Manager relies on a series of XML files that define a new target type to the system. A Management Plug-in consists of all files required to monitor a specific target from within Enterprise Manager.

---

---

**Important:** Much of Enterprise Manager is built around XML files. For this reason, it is essential that you are familiar with XML and related technologies. **This chapter assumes that you have a working knowledge of XML and DTDs.**

- **XML** (eXtensible Markup Language) - An XML document is a structured text document that conveys a particular kind of information.
  - **DTD** (Document Type Declaration) - A DTD provides the grammar for the XML files, thus describing what content is expected in each of its related XML files. When creating a new XML file, you need to carefully study its DTD to understand what content needs to be present in that file.
- 
- 

Once a Management Plug-in has been defined, you add it to a Management Plug-in Archive that is imported into Enterprise Manager. An archive encapsulates the one or more Management Plug-ins within a single jar file.

Creating a Management Plug-in consists of five stages:

1. Designing your Management Plug-in.
2. Developing the requisite target type definition files that make up the Management Plug-in.
3. Validating the target type definition.
4. Using the Enterprise Manager Command Line Interface (EM CLI), Package the target type definition files into a portable Management Plug-in Archive file.

5. Deploy the Management Plug-in throughout your Enterprise Manager environment.

## 2.1 Designing Your Management Plug-in

Before creating actual plug-in files, you need to define what parameters of the target type are required to accurately monitor and manage your new component. This involves:

- Identifying performance and configuration metrics that should be collected.
- Determining how often each metric should be collected. Oracle recommends that the collection frequency for any metric should not be less than once every five minutes.
- Based on customer-specific operational practices, specifying default warnings and/or critical thresholds on these metrics. Whenever a threshold is crossed, Enterprise Manager generates an alert, informing administrators of potential problems.

## 2.2 Developing Requisite Management Plug-in Files

Once you have determined how a particular type of target should be monitored, you are ready to begin creating the Management Plug-in files that define this particular target type to Enterprise Manager. To create a target type is to define to Enterprise Manager not only what a target is, but more importantly what should be monitored (metrics), and how monitoring should be performed.

### File Locations

The following table lists the key files used by Enterprise Manager for system configuration and extensibility. When a Management Plug-in is deployed to an Agent, the requisite files are copied to their respective directory locations.

**Table 2–1 Configuration File Locations**

Name	Directory Location	Purpose
target_type.xml (Required to define a target type.)	AGENT_HOME/sysman/admin/metadata/	Defines metrics to be collected/computed for the target type.
target_collections.xml (Required to define a target type.)	AGENT_HOME/sysman/admin/default_collection/	Defines metric collection parameters such as metric data collection intervals and default alert thresholds. Typically, the name of the metadata (target_type.xml) file and the default collection file are identical.
targets.xml	AGENT_HOME/sysman/emd/	Lists user-added target instances added to the Enterprise Manager framework. This file is generated and updated by Enterprise Manager and/or the Target Instance Installer utility. This file should not be edited manually.

DTD files associated with the XML configuration files can be found at the locations specified in [Table 2–2](#).

**Table 2–2 Document Type Declaration Files**

Name	Directory Location	Purpose
TargetMetadata.dtd	AGENT_HOME/sysman/admin/dtds/	Documentation Type Declaration (DTD) file associated with the target_type.xml file.
TargetInstance.dtd	AGENT_HOME/sysman/admin/dtds/	DTD associated with the targets.xml file.
TargetCollection.dtd	AGENT_HOME/sysman/admin/dtds/	DTD associated with the target_collections.xml files.

When you deploy a Management Plug-in, Enterprise Manager automatically copies all plug-in files to the appropriate locations. Conversely, when you are ready to update or remove a Management Plug-in, appropriate files are replaced in the case of a plug-in upgrade, or removed if a plug-in is undeployed.

---

**Important:** The Management Plug-in deployment mechanism does not permit overwriting of Oracle-supplied metadata files.

---

## 2.2.1 Target Definition Files

Adding a target type to the Enterprise Manager framework makes it possible for any instance of that target type to be monitored by an Oracle Management Agent once the target instance has been added to the framework from the Enterprise Manager Grid Control console. Every target type within the Enterprise Manager Grid Control framework must be unique. Defining a target type and making this information available to an Oracle Management Agent involves creating two XML files:

### Target Type Metadata File

The definition of a target type is composed primarily of the metrics you wish to expose. The file contains a list of all metrics collected for a particular target type, along with specifics on how to compute each metric.

### Target Type Default Collections File

Whether you want to regularly upload metric values to the Management Repository or check these values against specific conditions, you will need to define default collection intervals for each of the target metrics. This is also the file where you specify alert thresholds for each metric, enabling events to be triggered when violations occur. Users will be allowed to override the defaults, but the original collections file must be provided by the target provider.

The following steps take you through the process of creating the target type metadata and collection files, registering them with the Enterprise Manager framework, and finally adding instances of the new target type to the Enterprise Manager Grid Control console for monitoring.

#### 2.2.1.1 Creating the target type metadata file.

Target type metadata consists of the metrics you want to expose and the methods used to retrieve and compute those metrics. The target type metadata file tells the Oracle Management Agent what data to retrieve and how to obtain that data for this particular target type. Users can add a new target type by creating a new target metadata XML file and placing it under \$AGENT\_HOME/sysman/admin/metadata/.

---

---

**Note:** Manually placing the target metadata XML file in the AGENT\_HOME metadata directory is for development purposes only. The target metadata file is automatically placed in the metadata directory when the Management Plug-in is deployed to the Agent.

---

---

### Target Type Metadata File Naming Conventions

Although you can specify any name for a target type metadata file, Oracle recommends that users adding new target types adhere to Enterprise Manager naming conventions. This naming convention allows for file naming consistency in environments where similar products from multiple vendors are used. The target naming convention follows the form <vendor>\_<product category>. For example:

```
cisco_router.xml
oracle_apache.xml
bea_http.xml
cisco_slb.xml
f5_slb.xml
nortel_slb.xml
```

To account for cases where one product category contains multiple product offerings from a single vendor, Oracle recommends that you use the following file naming convention:

```
<vendor>_<category>_<brand name>
```

For example:

```
f5_slb_bigip.xml
f5_slb_3dns.xml
```

Enterprise Manager ships with a wide array of predefined target type metadata files that cover the most common target types. In situations where the predefined target metadata files do not fit exactly the types of targets you wish to monitor, you can either create a new target metadata file from scratch, or use one of the predefined metadata files as a template for defining a new target type, and then repackage them as a new Management Plug-in. The pre-defined metadata files are located in the \$AGENT\_HOME/sysman/admin/metadata directory.

### Anatomy of a Target Type Metadata File.

At the highest definition level, the target type metadata file is composed of four primary XML elements:

- TargetMetadata (top level)
  - Display (Oracle Internal Use)
  - Metric
  - InstanceProperties

Although this section briefly introduces XML elements and some of tag options used to define target metadata, for explicit definitions and other important usage information, see the TargetMetadata.dtd file located in the \$AGENT\_HOME/sysman/admin/dtds directory.

The following example shows the primary XML elements of the Net App Filer target type metadata file.



**Example 2-1 Target Type Metadata File**

```

<?xml version="1.0" ?>
<!DOCTYPE TargetMetadata SYSTEM "../dtds/TargetMetadata.dtd">

<TargetMetadata META_VER="2.2" TYPE="netapp_filer" CATEGORY_
PROPERTIES="IsClusterEnabled;IsSnapLicensed;Filer_Version">

<Display>
  <Label NLSID="netapp_filer">Network Appliance Filer</Label>
</Display>

<!-- *****
***** -->

<Metric NAME="Product" TYPE="TABLE">
  <Display>
    <Label NLSID="netapp_filer_info">Product Information</Label>
  </Display>
  <TableDescriptor>
    <ColumnDescriptor NAME="productCategory" TYPE="STRING">
      <Display>
        <Label NLSID="netapp_filer_product_category">Product Category</Label>
      </Display>
    </ColumnDescriptor>
    <ColumnDescriptor NAME="prodVersion" TYPE="STRING">
      <Display>
        <Label NLSID="netapp_filer_product_version">Version</Label>
      </Display>
    </ColumnDescriptor>
    <ColumnDescriptor NAME="prodId" TYPE="STRING">
      <Display>
        <Label NLSID="netapp_filer_product_id">Product ID</Label>
      </Display>
    </ColumnDescriptor>
    <ColumnDescriptor NAME="prodVendor" TYPE="STRING">
      <Display>
        <Label NLSID="netapp_filer_product_vendor">Vendor</Label>
      </Display>
    </ColumnDescriptor>
    <ColumnDescriptor NAME="prodModel" TYPE="STRING">
      <Display>
        <Label NLSID="netapp_filer_product_model">Model</Label>
      </Display>
    </ColumnDescriptor>
    <ColumnDescriptor NAME="firmwareVersion" TYPE="STRING">
      <Display>
        <Label NLSID="netapp_filer_firmware_version">Firmware
Version</Label>
      </Display>
    </ColumnDescriptor>
    <ColumnDescriptor NAME="prodVersionBucket" TYPE="STRING" COMPUTE_
EXPR="(prodVersion __contains 'NetApp Release 7.') ? '7.X':'6.X'"
TRANSIENT="TRUE">
      <Display>
        <Label NLSID="netapp_filer_prodVersionBucket">Product Version
Bucket</Label>
      </Display>
    </ColumnDescriptor>
  </TableDescriptor>
  <QueryDescriptor FETCHLET_ID="Snmp">

```

```

        <Property NAME="NAME" SCOPE="INSTANCE">NAME</Property>
        <Property NAME="hostname" SCOPE="INSTANCE">AgentHost</Property>
        <Property NAME="COMMUNITY" SCOPE="INSTANCE"
OPTIONAL="TRUE">CommunityString</Property>
        <Property NAME="TIMEOUT" SCOPE="INSTANCE" OPTIONAL="TRUE">Timeout</Property>
        <Property NAME="TABLE" SCOPE="GLOBAL">FALSE</Property>
        <Property NAME="OIDS" SCOPE="GLOBAL">1.3.6.1.2.1.1.2.0
                                                1.3.6.1.4.1.789.1.1.2.0
                                                1.3.6.1.4.1.789.1.1.3.0
                                                1.3.6.1.4.1.789.1.1.4.0
                                                1.3.6.1.4.1.789.1.1.5.0
                                                1.3.6.1.4.1.789.1.1.6.0

        </Property>
    </QueryDescriptor>
</Metric>
.
.
.
<InstanceProperties>

    <DynamicProperties NAME="ClusterConfig" FORMAT="ROW"
        PROP_LIST="IsClusterEnabled">
        <QueryDescriptor FETCHLET_ID="Snmp">
            <Property NAME="NAME" SCOPE="INSTANCE">NAME</Property>
            <Property NAME="hostname" SCOPE="INSTANCE">AgentHost</Property>
            <Property NAME="COMMUNITY" SCOPE="INSTANCE"
OPTIONAL="TRUE">CommunityString</Property>
            <Property NAME="TIMEOUT" SCOPE="INSTANCE"
OPTIONAL="TRUE">Timeout</Property>
            <Property NAME="TABLE" SCOPE="GLOBAL">FALSE</Property>
            <Property NAME="OIDS"
SCOPE="GLOBAL">1.3.6.1.4.1.789.1.2.3.1.0</Property>
        </QueryDescriptor>
        </DynamicProperties>

    <DynamicProperties NAME="VersionConfig" FORMAT="ROW"
        PROP_LIST="Filer_Version">
        <ExecutionDescriptor>
            <GetTable NAME="Product"/>
            <GetView NAME="Config" FROM_TABLE="Product">
                <Column NAME="prodVersionBucket"/>
            </GetView>
        </ExecutionDescriptor>
        </DynamicProperties>

    <DynamicProperties NAME="SnapMirrorConfig" FORMAT="ROW"
        PROP_LIST="IsSnapLicensed">
        <QueryDescriptor FETCHLET_ID="Snmp">
            <Property NAME="NAME" SCOPE="INSTANCE">NAME</Property>
            <Property NAME="hostname" SCOPE="INSTANCE">AgentHost</Property>
            <Property NAME="COMMUNITY" SCOPE="INSTANCE"
OPTIONAL="TRUE">CommunityString</Property>
            <Property NAME="TIMEOUT" SCOPE="INSTANCE"
OPTIONAL="TRUE">Timeout</Property>
            <Property NAME="TABLE" SCOPE="GLOBAL">FALSE</Property>
            <Property NAME="OIDS"
SCOPE="GLOBAL">1.3.6.1.4.1.789.1.9.19.0</Property>
        </QueryDescriptor>
        </DynamicProperties>

```

```

<InstanceProperty NAME="AgentHost" CREDENTIAL="FALSE" OPTIONAL="FALSE">
  <Display>
    <Label NLSID="netapp_filer_agent_host">Hostname or IP Address</Label>
  </Display>
</InstanceProperty>
<InstanceProperty NAME="CommunityString" CREDENTIAL="TRUE" OPTIONAL="TRUE">
  <Display>
    <Label NLSID="netapp_filer_community_string">SNMP Read Community String
(Default: public)</Label>
  </Display>
  public
</InstanceProperty>
<InstanceProperty NAME="Timeout" CREDENTIAL="FALSE" OPTIONAL="TRUE">
  <Display>
    <Label NLSID="netapp_filer_timeout">SNMP Timeout (Default: 5
seconds)</Label>
  </Display>
  5
</InstanceProperty>
.
.
.
</InstanceProperties>
</TargetMetadata>

```

From the target metadata file excerpt, you can see that the file consists of the following functional areas where you define your target metadata:

- TargetMetadata and Display
- Metric Name
- Instance Properties

### TargetMetadata and Display

The first lines after the header of the target definition file identify the target type. The following example is an excerpt from the Net App Filer target definition file.

#### **Example 2–2 TargetMetadata and Display XML Elements**

```

<?xml version="1.0" ?>
<!DOCTYPE TargetMetadata SYSTEM "../dtds/TargetMetadata.dtd">
<TargetMetadata META_VER="2.2" TYPE="netapp_filer" CATEGORY_
PROPERTIES="IsClusterEnabled;IsSnapLicensed;Filer_Version">
<Display>
  <Label NLSID="netapp_filer">Network Appliance Filer</Label>
</Display>

```

These lines define the basic specifications: metadata version (META\_VER="2.2"), target type (TYPE="netapp\_filer"), the target's NLS Identifier (NLSID="netapp\_filer") and display name text ("Network Appliance Filer"). Metadata versioning allows different versions of the same target type metadata to exist concurrently within the managed environment (one metadata version per Management Agent). You should change the metadata version each time you update the target metadata file. Typically a target type consists of the company name followed by the product name. The "Display" element is used internally by Oracle for translation purposes and is not required when defining new target types. "Display" elements are used with various elements in the target metadata for both readability and internationalization.

**Metric NAME**

The content of the target type metadata file is devoted primarily to metric definitions. As a matter of practice, Oracle recommends that you specify at least two metrics for any target type:

- Metric to monitor target availability (Required for all target types.)
- Metric to monitor target performance (Optional but recommended.)

For target availability to show up correctly on the default target home page, Oracle requires the target metadata file to define a metric with NAME="Response" that contains a column with NAME="Status" and the default collection file must define a critical condition on the "Status" column that represents the target being up or down.

Shown in the example below, the Response metric, which monitors target availability, is required for all target types. The Load metric is used to determine a target's performance level.

The following example shows the "Response" metric definition for a Network Appliance Filer target type.

**Example 2-3 Response Metric**

```
<Metric NAME="Response" TYPE="TABLE">
  <Display>
    <Label NLSID="netapp_filer_resp">Response</Label>
  </Display>
  <TableDescriptor>

    <ColumnDescriptor NAME="Status" TYPE="NUMBER">
      <Display>
        <Label NLSID="netapp_filer_resp_status">Status</Label>
      </Display>
    </ColumnDescriptor>

  </TableDescriptor>
  <QueryDescriptor FETCHLET_ID="Snmp">
    <Property NAME="NAME" SCOPE="INSTANCE">NAME</Property>
    <Property NAME="hostname" SCOPE="INSTANCE">AgentHost</Property>
    <Property NAME="COMMUNITY" SCOPE="INSTANCE"
OPTIONAL="TRUE">CommunityString</Property>
    <Property NAME="TIMEOUT" SCOPE="INSTANCE"
OPTIONAL="TRUE">Timeout</Property>
    <Property NAME="PINGMODE" SCOPE="GLOBAL">TRUE</Property>
    <Property NAME="OIDS" SCOPE="GLOBAL">1.3.6.1.4.1.789.1.1.3.0</Property>
  </QueryDescriptor>
</Metric>
```

Because the "Response" metric is of type "TABLE", it requires a "TableDescriptor" and associated "ColumnDescriptor". Note that all levels of the metric, which includes table and column definitions, are given names, types, and display labels. As mentioned earlier, these elements and usage instructions can be found in the TargetMetadata.dtd file.

For each metric, you must also specify how to obtain the data to fill this structure. This is accomplished using fetchlets—mechanisms that take parameters, query a specified target, and then return some values from the target. Enterprise Manager provides a wide array of fetchlets to meet most data retrieval needs. See [Chapter 8, "Fetchlets"](#) and [Chapter 5, "Management Using Web Services"](#) for more information on fetchlets and available fetchlet types.

A fetchlet associated with the metric is declared through the "QueryDescriptor" tag. The following example shows the QueryDescriptor used for the Microsoft SQL Server Response metric.

#### **Example 2-4 Query Descriptor Usage**

```
<QueryDescriptor FETCHLET_ID="OSLineToken">
<Property NAME="emdRoot" SCOPE="SYSTEMGLOBAL">emdRoot</Property>
<Property NAME="command" SCOPE="GLOBAL">%emdRoot%/bin/nmefwmi</Property>
<Property NAME="ENVWBEM_WQL" SCOPE="GLOBAL">select name,pathname,processid,state
from win32_service where pathname like '%sqlservr.exe%' or pathname like
'%sqlagent%'</Property>
<Property NAME="ENVWBEM_WQL_COLUMN_ORDER"
SCOPE="GLOBAL">name,pathname,processid,state</Property>
<Property NAME="startsWith" SCOPE="GLOBAL">em_result</Property>
<Property NAME="delimiter" SCOPE="GLOBAL">|</Property>
<Property NAME="NEED_CONDITION_CONTEXT" SCOPE="GLOBAL">TRUE</Property>
</QueryDescriptor>
```

The "FETCHLET\_ID" label is used to specify which fetchlet to use for data retrieval. In the example above, the "Response" metric uses the "OSLineToken" fetchlet. What follows are the definition of required system parameters used by the fetchlet.

The "oraTclHome", "perlBin", and "scriptsDir" properties enable the correct bindings in the "command" property. The remaining properties are "OSLineToken" fetchlet input parameters. See [Chapter 8, "Fetchlets"](#) for information on parameters used by each fetchlet.

The "scope" property defines where the property value is to be obtained. The following scope options are available:

- SCOPE="GLOBAL" Obtain the property value from other variables defined within the current target type metadata file. This includes constants, such as the "|" shown in the QueryDescriptor example.
- SCOPE="INSTANCE" Obtain the property value from instance properties.
- SCOPE="ENVxxx" Obtain the property value from an environment variable "xxx".
- SCOPE="SYSTEMGLOBAL" Obtain the property value from the emd.properties file located in the \$AGENT\_HOME/sysman/config directory.
- SCOPE="USER" Obtain the property value from the collector or user.

#### **Instance Properties**

The "InstanceProperties" descriptor is a required top-level specification that defines what properties an administrator must specify in the Enterprise Manager Grid Control console when adding a new target instance of this particular target type.

Although the InstanceProperties section can be defined at various locations within the target type metadata file, Oracle recommends defining this section at the very end of the file for consistency. Instance properties defined in the target type metadata file are resolved to values specified for these instance properties in the targets.xml file. Shown below are some of the instance properties for a Microsoft SQL Server target type.

#### **Example 2-5 Instance Properties**

```
<InstanceProperties>
<InstanceProperty NAME="ms_sqlserver_host" CREDENTIAL="FALSE" OPTIONAL="FALSE">
<Display>
```

```

<Label NLSID="ms_sqlserver_host">SQL Server host</Label>
</Display>
</InstanceProperty>
<InstanceProperty NAME="ms_sqlserver_servername" CREDENTIAL="FALSE"
OPTIONAL="FALSE">
<Display>
<Label NLSID="ms_sqlserver_servername">SQL Server name</Label>
</Display>
</InstanceProperty>
<InstanceProperty NAME="url" CREDENTIAL="FALSE" OPTIONAL="FALSE">
<Display>
<Label NLSID="connectString_iprop">JDBC URL</Label>
</Display>
</InstanceProperty>
.
.
.
</InstanceProperties>

```

As an example, when an Oracle Database Instance target type is added to the Enterprise Manager framework, specific information about that target instance is added to the targets.xml file. In the example below, we can see that the InstanceProperty labels such as OracleHome, Port, SID defined in the Oracle Database Instance target type metadata file are resolved to instance-specific values in the targets.xml file.

#### **Example 2-6 Target Instance Information for an Oracle Database**

```

<Target TYPE="oracle_database" NAME="emrep.us.oracle.com">
<Property NAME="OracleHome" VALUE="/scratch/OracleHomes/db10g"/>
<Property NAME="UserName" VALUE="f8a6d72528517730" ENCRYPTED="TRUE"/>
<Property NAME="MachineName" VALUE="stacb11.us.oracle.com"/>
<Property NAME="Port" VALUE="1521"/>
<Property NAME="SID" VALUE="emrep"/>
<Property NAME="ServiceName" VALUE="emrep.us.oracle.com"/>
<Property NAME="password" VALUE="7536910dcf78eaad" ENCRYPTED="TRUE"/>
</Target>

```

#### **2.2.1.2 Validate your new target type definitions.**

It is always a good idea to test your new target type definitions using the metric browser. The metric browser is a development utility that is an integral part of the Oracle Management Agent. As a subsystem of the Agent, it allows you to quickly access the metric values for targets monitored by the Agent without the overhead of a Management Repository or other components of the Enterprise Manager framework.

#### **Activating the Metric Browser**

To configure the Oracle Management Agent's metric browser for debugging metrics without the Enterprise Manager Grid Control console:

1. Uncomment the enableMetricBrowser line in the \$AGENT\_HOME/sysman/config/emd.properties file:

```

#enableMetricBrowser=True

```

change to

```

enableMetricBrowser=True

```

2. Make sure the Oracle Management Agent is stopped.

```
emctl stop agent
```

3. Restart the Management Agent.

```
emctl start agent
```

4. Open the emd.properties file and check the EMD\_URL line.

It will have the following format:

```
EMD_URL=http://<host>:<port>/emd/main
```

### Adding a Target Instance to the targets.xml File

Before you can view your target metrics in the metric browser, you need to add an actual instance of your new target type to the targets.xml file without re synchronizing the Management Agent with the Management Service. To do this:

1. Create a new XML file containing the instance information for a specific target. The contents of this XML file must conform to the \$AGENT\_HOME/sysman/admin/dtds/TargetInstance.dtd file. The content for this file consists of the actual values specified for the instance properties you defined in the target type metadata file. For example, given the instance properties defined in a particular target metadata file, you create an XML file called "target2add.xml" that contains the following:

#### Example 2-7 target2add.xml File

```
<Target TYPE="oracle_listener" NAME="listener" VERSION="1.0">
<Property NAME="ListenerOraDir" VALUE="/myOH/oracle/network/admin"/>
<Property NAME="LsnrName" VALUE="LISTENER"/>
<Property NAME="Machine" VALUE="mymachine.us.oracle.com"/>
<Property NAME="OracleHome" VALUE="/myOH/oracle"/>
<Property NAME="Port" VALUE="15091"/>
</Target>
```

2. Use the command line instance installer to add the contents of the instance information file to the Management Agent's targets.xml file and reload the instance data. For the examples below, Management Agent home is located in /em/ and the instance information file is target2add.xml located in the /tmp/ directory.

Add the contents of your new target information file to the targets.xml file.

```
emctl config agent addtarget <file location> <Agent Home location>
```

Example:

```
emctl config agent addtarget /tmp/target2add.xml /em/
```

Verify that your target information was added correctly:

```
emctl config agent listtargets <emloc>
```

Example:

```
emctl config agent listtargets /em/
```

Reload the modified target information. This operation is required in order for the new target instance to appear in the metric browser.

```
emctl reload
```

## Looking at Your Metrics

Once the target instance has been added to the targets.xml file and the new information has been reloaded, you can view available targets and metrics through the metric browser. Access the following URL using any web browser

```
http://<host>:<port>/emd/browser/main
```

To find the port number used by the Management Agent, open the emd.properties file and search for the EMD\_URL line.

### 2.2.1.3 Creating a Default Collection File

Defining your target type metrics allows you to view the data belonging to that target instance at any time. However, you may also wish to collect the values of some of your metrics to analyze the variations over some period of time. You set up metric collection intervals for your target instances by defining a default collections file for the specific target type.

---

**Note:** Full definitions and usage instructions for descriptor tags used in the collection file can be found in the \$AGENT\_HOME/sysman/admin/dtds/TargetCollection.dtd file.

---

**2.2.1.3.1 Default Collection File** Located in \$AGENT\_HOME/sysman/admin/default\_collection, the default collection file (XML) for a target type specifies the metrics to be collected and sent to the Management Repository, as well the frequency of these scheduled collections. Although you can use any name for a default collections file, it is recommended that you use a naming convention that makes it easy to associate the collection file with the corresponding target type metadata filename, such as using a filename identical to the target type metadata filename.

The following example is an excerpt from the Microsoft SQL Server default collections file.

**Example 2–8 Microsoft SQL Server Default Collections File**

```
<!DOCTYPE TargetCollection SYSTEM "../dtds/TargetCollection.dtd" []>
<TargetCollection TYPE="microsoft_sqlserver_database">
<CollectionItem NAME="Response">
<Schedule>
<IntervalSchedule INTERVAL="5" TIME_UNIT="Min" />
</Schedule>
<Condition COLUMN_NAME="Status" CRITICAL="Running" OPERATOR="NE" />
</CollectionItem>

<CollectionItem NAME="SQLServer" UPLOAD_ON_FETCH="TRUE" CONFIG="TRUE">
<Schedule OFFSET_TYPE="INCREMENTAL">
<IntervalSchedule INTERVAL="5" TIME_UNIT="Min" />
</Schedule>
<MetricColl NAME="MSSQL_SQLServer" />
<MetricColl NAME="MSSQL_RegistrySetting" />
</CollectionItem>
.
.
.
<!-- Buffer cache hit ratio that is lower than 90% triggers warning and 80%
triggers alert -->
<!-- Cache hit ratio that is lower than 80% triggers warning and 70% triggers
alert -->
```



```

<CollectionItem NAME="MSSQL_MemoryStatistics">
<Schedule>
<IntervalSchedule INTERVAL="10" TIME_UNIT="Min" />
</Schedule>
<Condition COLUMN_NAME="buffer_cache_hit_ratio" WARNING="90" CRITICAL="80"
OPERATOR="LE" />
<Condition COLUMN_NAME="cache_hit_ratio" WARNING="80" CRITICAL="70" OPERATOR="LE"
/>
</CollectionItem>

<CollectionItem NAME="MSSQL_Alert">
<Schedule>
<IntervalSchedule INTERVAL="20" TIME_UNIT="Min" />
</Schedule>
</CollectionItem>

<CollectionItem NAME="MSSQL_LastDatabaseBackup">
<Schedule>
<IntervalSchedule INTERVAL="10" TIME_UNIT="Min" />
</Schedule>
</CollectionItem>

</TargetCollection>

```

For every target instance, the data collection intervals for the four metrics would be as follows:

- **Response:** Collected every 5 minutes.
- **Load:** Collected every 15 minutes.
- **memPool:** Collected every 15 minutes.
- **Interfaces:** Collected every 15 minutes.

If the threshold is not set in the collection file, administrators will not be able to edit/add the column threshold value from the Enterprise Manager console at a later point. To allow an administrator to change a threshold that does not have a default value, you can add a "NotDefined" entry for a specific threshold. For example:

```
<Condition COLUMN_NAME="db_session_osuser" CRITICAL="NotDefined"/>
```

**2.2.1.3.2 Target Instance-Specific Collection Schedules** Under certain circumstances, you may not want to have all target instances use the same collection schedule. To specify that different instances of a target type have different collection schedules, you can add additional collections files (for specific target instances) to \$AGENT\_HOME/sysman/emd/collections.

The following example shows a situation where we want two specific target instances ("Simple Server Alpha" and "Simple Server Beta") to have different "Response" and "Load" collection schedules. The two collection files to be added to the collections directory would contain the following:

**Example 2-9 Default Collection File for Simple Server Alpha**

```

<TargetCollection NAME="Simple Server Alpha" TYPE="simple_server">
  <CollectionItem NAME="Response">
    <Schedule>
      <IntervalSchedule INTERVAL="10" TIME_UNIT="Min"/>
    </Schedule>
  </CollectionItem>
</TargetCollection>

```

**Example 2–10 Default Collection File for Simple Server Beta**

```
<TargetCollection NAME="Simple Server Beta" TYPE="simple_server">
  <CollectionItem NAME="Load">
    <Schedule>
      <IntervalSchedule INTERVAL="30" TIME_UNIT="Min"/>
    </Schedule>
  </CollectionItem>
</TargetCollection>
```

Here we are collecting results from Simple Server Alpha's "Response" every 10 minutes and Simple Server Beta's "Load" every 30 minutes.

**2.2.1.4 Adding SYSTEM Reports (Optional)**

Developing a Management Plug-in allows you to add new SYSTEM reports that are associated with the target type(s) defined by your Management Plug-in. SYSTEM reports cannot be edited or deleted from the Enterprise Manager console. All out-of-box reports supplied with Information Publisher are SYSTEM reports.

To add report definitions to your Management Plug-in, you simply create a report definition file for inclusion with the plug-in. A report definition file consists of a conventional PL/SQL block that specifies pertinent information to extract from the Management Repository and the report elements used to format and display that data. You may define multiple reports for a given target type. For more information on the Information Publisher PL/SQL API and creating a plug-in report definition file, see [Chapter 4, "Adding Reports"](#).

**2.2.1.5 Adding Related Links to Target Home Pages**

If reports are defined as part of a Management Plug-in, and at least one of the reports is registered to be shown in the context of the target default home page (it is registered using the "add\_report" PL\*SQL function) then one or more related links may be added to the target default humped using the process described in this section.

Related Links are defined as dynamic instance properties in the target metadata file using the "DynamicProperties" tag with the following parameters:

- NAME="*<any name>*"
- PROP\_LIST="*RelatedLink\_Name\_#;RelatedLink\_Dest\_#[;...]*"
  - Where # can be any positive integer (Example: 1, 2, or 3)
  - PROP\_LIST must be EXACTLY in the format specified here (including bumpy case). Otherwise, the related links will not show up on the target default home page.

Also, a "QueryDescriptor" tag block that retrieves the number of "Name/Dest" pairs listed in PROP\_LIST is required. The "RelatedLink\_Name\_#" appears as the hyperlinked text. The "RelatedLink\_Dest\_#" is the link destination (URL or Javascript).

[Example 2–11](#) is an excerpt from a target metadata file that contains a "DynamicProperties" block defining three related links that will appear on the default target home page.

**Example 2–11 Related Links**

```
<InstanceProperty NAME="EMPLOYEE_ID" CREDENTIAL="FALSE" OPTIONAL="FALSE">
  <Display>
    <Label NLSID="EMPLOYEE_ID_iprop">Employee ID</Label>
```

```

</Display>
</InstanceProperty>
  <DynamicProperties NAME="Links" PROP_LIST="RelatedLink_Name_1;RelatedLink_Dest_
1;RelatedLink_Name_2;RelatedLink_Dest_2;RelatedLink_Name_3;RelatedLink_Dest_3"
FORMAT="ROW">
  <QueryDescriptor FETCHLET_ID="OSLineToken">
    <Property NAME="id" SCOPE="INSTANCE">EMPLOYEE_ID</Property>
    <Property NAME="command" SCOPE="GLOBAL">
      %perlBin%/perl %scriptsDir%/emx/oracle/getLinks.pl %id%
    </Property>
    <Property NAME="startsWith" SCOPE="GLOBAL">em_result=</Property>
    <Property NAME="delimiter" SCOPE="GLOBAL">|</Property>
  </QueryDescriptor>
</DynamicProperties>
</InstanceProperties>

```

Note that in [Example 2–11](#) the Perl script “getLinks.pl” is used to return the three links. The content of this script is shown in [Example 2–12](#).

#### **Example 2–12 Script getLinks.pl**

```

my $id = $ARGV[0];

my $name1 = "Employee: MY-PC";
my $link1 = "javascript:void window.open('http://my-pc.us.oracle.com', 'aria',
'resizable=1,menubar=1,toolbar=1,titlebar=1,status=1,scrollbars=1')";
my $name2 = "Employee: Hierarchy";
my $link2 = "javascript:void
window.open('http://employee.us.oracle.com:7777/pls/oracle/f?p=8000:3:720062688223
0853285::NO::PERSON_ID:$id', 'employee',
'resizable=1,menubar=1,toolbar=1,titlebar=1,status=1,scrollbars=1')";
my $name3 = "OTN";
my $link3 = "javascript:void window.open('http://otn.oracle.com', 'otn',
'resizable=1,menubar=1,toolbar=1,titlebar=1,status=1,scrollbars=1')";

print "em_result=$name1|$link1|$name2|$link2|$name3|$link3\n";

```

Because “getLinks.pl” is a support script, you add it to the Management Plug-in as a separate file using the EM CLI “add\_mp\_to\_mpa” verb with the argument:

```
-file="MONITORING_SCRIPT:<path_to>/getLinks.pl"
```

Be very careful with the “javascript” syntax you use. Only some of the syntax is common to the popular browsers. For example, the second argument to “window.open” is a window name used for internal reference to the window and it must not contain any spaces.

#### **2.2.1.6 Additional Scripts and Binaries**

In addition to the target metadata, collection, and report definition files, you may also include specialized scripts and/or binary files that are used to include obtain status or metric data from your custom targets. An example is the Perl script “getLinks.pl” mentioned in the previous section.

## **2.2.2 Ensuring Accurate XML**

Successfully creating target type metadata and collection files requires valid XML code. To assist with writing valid XML, Enterprise Manager provides ILINT, a development tool you can use to validate the XML used to define your code as you

develop your Management Plug-in. For more information on ILINT, see [Chapter 3, "Validating XML"](#).

## 2.2.3 Development Guidelines

When developing target type definition files for new Management Plug-ins, special consideration must be paid to the way in which you want a particular target type to be monitored. How a target type is monitored can greatly affect Enterprise Manager performance. General guidelines for defining target metadata and collections should be followed in order to optimize system performance.

### 2.2.3.1 Guidelines for Defining Target Metadata

Metadata is data about data. Generically, the term refers to any data used to aid in the identification, description and location of a network entity. Target metadata for an Enterprise Manager target consists of the metrics a user wants to expose and the methods used to compute those metrics.

**2.2.3.1.1 Metadata Version** Whenever the target metadata changes, the metadata version (`meta_ver`) should be incremented.

**2.2.3.1.2 Real-Time Only Metrics** Performance metrics can be classified into metrics that need to be computed to track performance trends and others that are more useful to drill down to get the details at a particular point in time. Real-time only metrics include those that need contextual information to return detailed information about a particular subset of the system, e.g., a specific tablespace to diagnose further.

**2.2.3.1.3 Choice of Key Columns** A key column in a metric is used in the management repository to trend performance data on an axis, e.g., the tablespace usage per database tablespace. An inappropriately chosen key column can result in too much collected data within the management repository. For instance, using the process ID in a Processes metric to upload to the repository.

You can have no key columns, but the query descriptor must return a single row.

**2.2.3.1.4 Transient Columns** In some cases, metric columns can be used to compute the values of other more interesting metric columns. In the cases where the original columns are not interesting for trending, these can be marked transient so that they are not uploaded to the repository and waste space.

**2.2.3.1.5 Metrics and Microsoft Windows** When creating metrics for custom targets, it is important to take into account the cost (CPU usage) of creating additional operating system (OS) processes. This is especially true for systems running Microsoft Windows where process creation is much more CPU intensive compared to UNIX-based systems such as Linux or Solaris. The percentage CPU utilization increases linearly with creation of child processes. To minimize process creation, avoid executing OS programs or commands from metric collection scripts. For example, when writing Perl scripts, avoid using the `system` function or backticks (```) to execute an OS command.

**2.2.3.1.6 Target Properties (Static Versus Dynamic)** Target properties are named values that can be used for computing the metrics of the target, or for display in the home page of the target. The list of target properties is specified in the metadata to allow data driven user interfaces to register targets, and for the Oracle Management Agent to validate that a target instance is complete.

- **Static Instance Properties:** These are properties whose values need to be specified for a target in the `targets.xml` entry for the target. An instance property can be

marked optional if the target declaration is considered complete even without the specification of the property. The metadata specification of a target property can also provide a default value for use in a configuration user interface.

- **Dynamic Instance Properties:** The Oracle Management Agent also allows for target instance properties to be "computed". Such properties are computed using a QueryDescriptor very similar to the ones used in metrics.

Use of dynamic properties reduces the work involved in configuring a target by allowing certain properties to be computed rather than requiring the user to correctly specify their values (for example, the "Version" property of a database can be reliably computed given addressing information).

The Oracle Management Agent allows for the fact that the target needs to be up for the successful computation of these dynamic properties by recomputing the properties each time a target bounce is detected (each time the target status changes to "Up").

**2.2.3.1.7 Metrics** The metric concept, as it pertains to the Oracle Management Agent, can be used to denote configuration and performance information.

- **Configuration Metrics:** Configuration metrics collect data similar to target properties that denote the configuration of the target. This information is periodically refreshed and can be used to track changes in the setup of a target. The collection interval on such metrics is typically on the order of about 24 hours.
- **Performance Metrics:** Performance metrics are used to track the responsiveness of a target. These metrics are typically collected more often than configuration metrics though the interval of some performance metrics may vary widely from those of others. Also, performance metrics usually ship with thresholds that are the basis of performance alerts for the target.

A required metric for all targets is the "Response" metric consisting of a "Status" column with a condition on it. This metric is used to track the availability of the target.

### Metric Naming Conventions

The conventions used in naming your metrics are extremely important because many areas of the Enterprise Manager user-interface are data-driven. For example, actual metric column labels and key values can be part of the page title, instruction text, or column headings. Specifically, these elements would appear on the Metric Details page, Edit Metric Threshold page, Notification Rules page, and other pages of the Enterprise Manager user-interface. For this reason, Oracle recommends the following metric naming conventions.

- All metric column names (labels) must be unique within a given target type and version, and easily understood by the user (metric units used as needed).  
Example: Tablespace Usage (%)
- All metric column names (labels) should be self-explanatory without dependence on the metric name.  
Example: Table Space Used (%)
- Key Column names should be self-explanatory. Key Column names are used when specifying metric thresholds or setting notifications. The following format should be used: all<key value name> objects  
Example: all (tablespace) objects

- Short names (up to 20 characters) associated with the metric column should be both clear and translatable.
- Across target versions, the same columns should use the same labels. This ensures columns, such as metric columns and short names, have the same NLS IDs across different target versions.

### 2.2.3.2 Guidelines for Defining Collections

Collections are the mechanism by which the Oracle Management Agent periodically computes the metrics of a target and uploads the data to the Management Repository. The most important thing to keep in mind when creating the collections for a target type is to avoid overburdening the Management Repository with excess data. In a large enterprise with hundreds of Oracle Management Agents and thousands of targets, the key to scalability is to limit the amount of data collected about a target that is uploaded to the repository. This is especially important since raw data is maintained for 24 hours - rollup benefits only accrue beyond that point.

**2.2.3.2.1 Alert Message Guidelines** Alert messages tell the user when something is wrong. These messages should also assist the user in solving the problem. We recommend following these content guidelines when writing alert messages:

- Alert messages should be meaningful. Avoid using terse, ambiguous messages unless the message is only applicable to the metric.
- Target down messages should, in addition to indicating that the target is down, include information indicating possible reasons why the target may be down.
- Error codes/messages should be included whenever possible.
- When appropriate, include information telling the user how to resolve or diagnose the problem.

**2.2.3.2.2 Metric Evaluation Order** It is important to pay attention to metric evaluation order so as to avoid metric collection failures. For example, the *Response* metric should be evaluated first in order to prevent a collection failure when a target is down. The Oracle Management Agent will evaluate metrics based on the order they are listed in the collection XML file.

---

---

**Note:** Programmatic logic of the Oracle Management Agent distributes the metric evaluations so that each evaluation is separated by approximately 10 seconds.

---

---

**2.2.3.2.3 Collection Frequency** In general, there is almost never a good reason to collect information at intervals smaller than 5 minutes. In the rare case where data variations occur at a smaller granularity and administrators need to be notified sooner, the Oracle Management Agent provides the capability to use a small collection interval to compute the metrics and threshold information while still only uploading data once in every *n* computation cycles.

**2.2.3.2.4 Controlling Number of Rows** Some metrics can result in the creation of a large number of rows in a Management Repository table. In some cases, only a subset of these rows may need to be uploaded to the repository. The Oracle Management Agent allows the specification of filter conditions that can be used to find rows to skip uploading. Also, a "limit\_to" clause can be used on metrics that return sorted metric data to upload only the first *n* rows to the repository.

## 2.3 Creating a Management Plug-In Archive

Once you have created the Management Plug-in files, the next step is to create a Management Plug-in Archive (MPA). The MPA plays an important role at various stages of the Management Plug-in lifecycle. It serves the following functions:

- As a transport mechanism between your development environment and the Enterprise Manager framework.
- As a transport mechanism between different Enterprise Manager installations.
- As a container for the Management Plug-in. A MPA may contain multiple Management Plug-ins.

A Management Plug-in is created by adding the files previously discussed to an MPA using the Enterprise Manager Command Line Interface (EM CLI). Each call to the EM CLI adds another unique Management Plug-in to the MPA. For each Management Plug-in, the EM CLI allows you to specify a base version of the Management Agent that the plug-in is expected to work against and a base version that the Oracle Management Service must be for the plug-in to be imported into the Management Repository. To create a MPA, perform the following

1. Open a terminal window on a machine where the EM CLI client is installed.
2. At the command prompt issue the `add_mp_to_mpa` verb. The following example shows the verb parameters that must be supplied. For more information about the `add_mp_to_mpa` verb, see the command line help.

### **Example 2–13 Using the EM CLI to Create a Management Plug-in Archive**

```
emcli add_mp_to_mpa
  -mpa="/my_dir/my_new_type.jar"
  -mp_version="2.0"
  -ttd="/my_dir/ttd/new_type.xml"
  -dc="/my_dir/dc/new_type.xml"
  -file="REPORT_DEFINITION:/my_dir/report1.sql
  -file="REPORT_DEFINITION:/my_dir/report2.sql
  -file="MONITORING_SCRIPT:/my_dir/script1.pl
  -file="MONITORING_SCRIPT:/my_dir/script2.pl
  -file="MONITORING_BINARY:/my_dir/bin1
  -func_desc="Management Plug-in to define target type new_type"
```

Briefly, the verb options are:

- **mpa**  
The name of the Management Plug-in Archive where the Management Plug-in is to be added.
- **mp\_version**  
The version of the Management Plug-in to be created. The Management Plug-in version should be incremented whenever any of the files in the management Plug-in are changed.
- **ttd**  
The explicit path of the target type metadata file.
- **dc**  
The explicit path of the default collection file.
- **oms\_version**

The minimum OMS version that is compatible with this Management Plug-in.

- **file**

The type and path of the other Management Plug-in files to be added. The following types are supported:

- MONITORING\_BINARY
- POLICY\_DEPLOY
- POLICY\_UNDEPLOY
- MONITORING\_SCRIPT
- REPORT\_DEFINITION

- **func\_desc**

The functional description for the Management Plug-in. This description appears in the Enterprise Manager console once the plug-in has been imported.

- **req\_desc**

The Requirements description of the Management Plug-in. This description appears in the Enterprise Manager console and specifies any plug-in deployment requirements.

## 2.4 Uploading the Management Plug-in Archive into Enterprise Manager

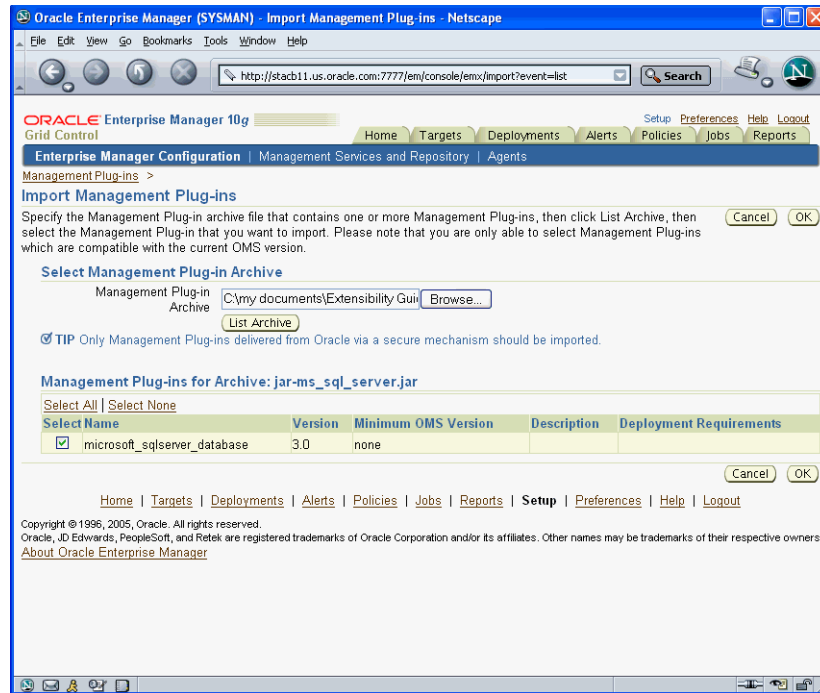
After using the EM CLI to create the Management Plug-in Archive, you are ready to upload the Management Plug-in Archive file to Enterprise Manager. Uploading the archive allows you to view all available plug-ins contained in the archive. You can then select which plug-ins you want to import into Enterprise Manager. You must have Super Administrator privileges in order to add Management Plug-ins to the system.

To upload a Management Plug-in Archive:

1. From the Enterprise Manager console, click Setup.
2. Click Management Plug-ins from the navigation bar at the left.
3. Click Import. The Import Management Plug-ins page displays.
4. In the Select Management Plug-in Archive section, specify the Management Plug-in Archive file.
5. Click List Archive to view Management Plug-ins contained within the archive.
6. Select the Management Plug-ins you want to import and click OK.

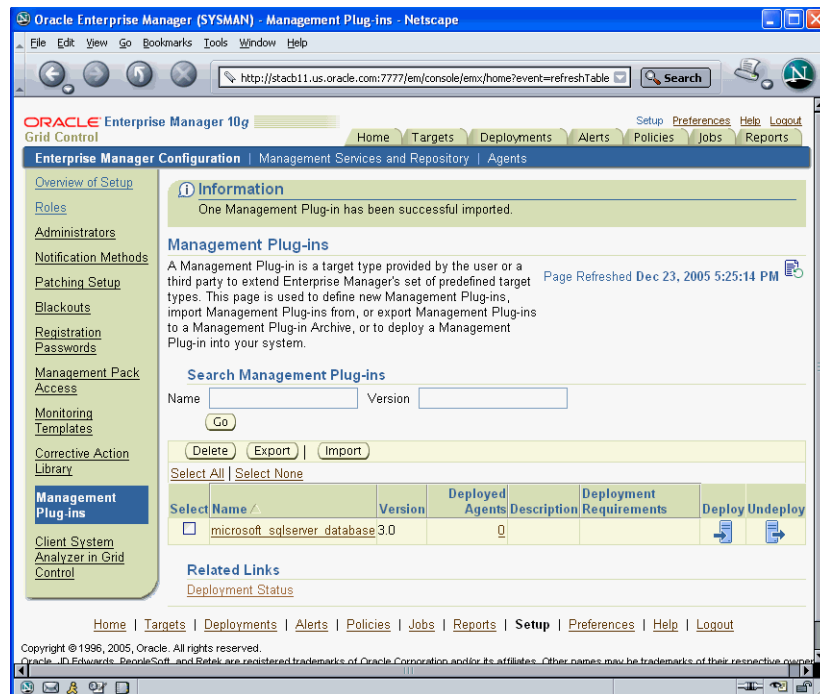


Figure 2–1 Import Management Plug-ins Page



Upon successful import, the plug-in appears in the Management Plug-in list.

Figure 2–2 Management Plug-in



At this point, the Management Plug-ins have been extracted from the archive file and imported into the Management Repository. The Management Plug-in is now ready to be deployed to Management Agents within your Enterprise Manager environment. To

deploy your plug-in, click the Deploy icon. Enterprise Manager guides you through a simple deployment process.

## 2.5 Adding a Target Instance

After a Management Plug-in has been deployed to an Agent, you are ready to add new target instances of the type defined by your Management Plug-in. When you add a target instance, monitoring and administrative functionality is automatically extended to that target. To add a target instance:

1. From the Monitored Targets section of the Management Agent home page, choose the target type defined by your plug-in from the Add drop-down menu and click Go. The Add target page appears.
2. Enter the requisite target properties and click OK. The newly added target appears in the Agent's Monitored Targets list.

A default target home page is provided that supplies requisite information about the target as shown in Figure 2-3.

## 2.6 Viewing Results

From a default target home page, a user can drill down to specific metrics that have been defined for the target type.

The following figure shows a target home page for a Microsoft SQL Server instance.

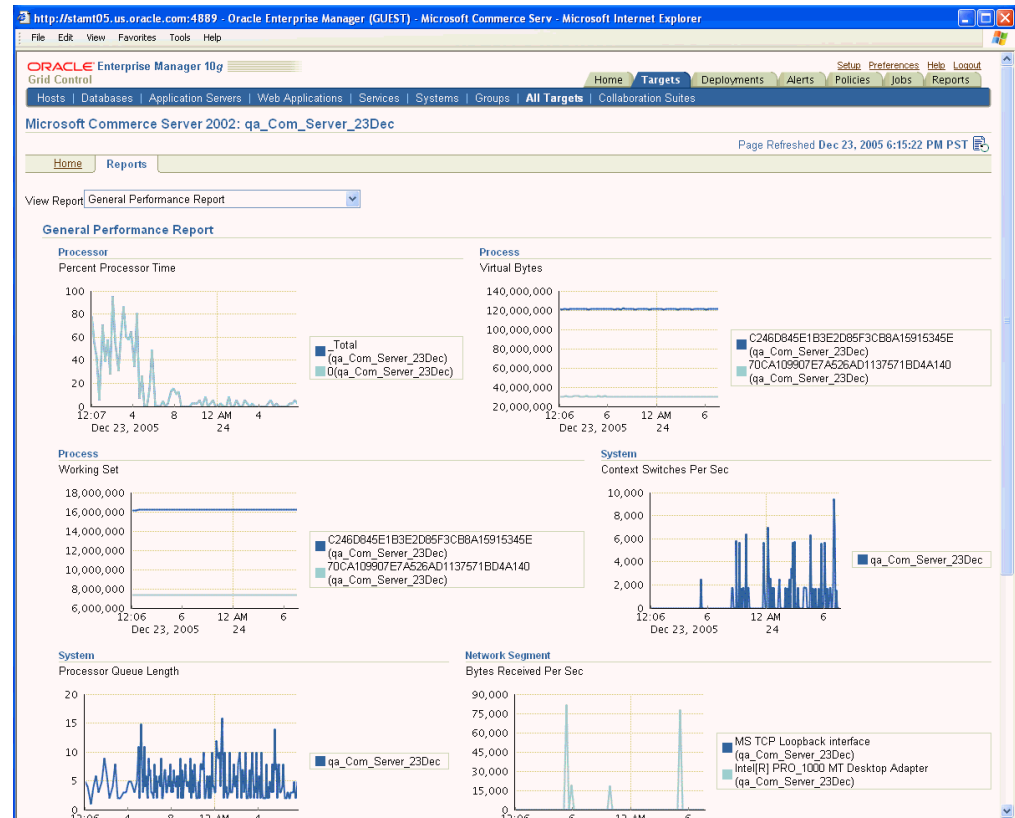
**Figure 2-3 Microsoft Commerce Server Target Home Page**



Because report definitions were included with this Management Plug-in, an additional Reports subtab appears with the home page. As shown in the next figure, adding reports can greatly improve the monitoring capabilities of your Management Plug-in.

For more information about creating Management Plug-in report definitions, see [Chapter 4, "Adding Reports"](#).

**Figure 2–4 Microsoft SQL Server Reports Subtab**



## 2.7 Troubleshooting Management Plug-ins

On occasion, you may encounter errors when when managing Management Plug-ins. The most common problems are as follows:

- **Deployment fails. "Agent Unreachable"**  
Ensure that the Management Agent is up and running.  
Log in as "root" and execute the root.sh script at the Management Agent.
- **Undeploy fails. "Agent Unreachable"**  
Ensure that the Management Agent is up and running.
- **Unable to delete the Management Plug-in.**  
The Management Plug-in must be undeployed from ALL Agents to which it was previously deployed. Note that this includes any Agents that are currently down.
- **Reports do not show any data.**  
Depending on the collection interval settings, it may take time for the Agent to collect the metric data for the target. Management views used by the reports can also determine when data is available, as would be the case if queries against the Management Repository are performed against rolled up data.



---

---

## Validating XML

Successfully defining new Management Plug-in target definition files relies on accurate XML: It is essential that your Management Plug-in's target metadata and collection files be syntactically and structurally correct. To assist you with developing your XML files, Enterprise Manager supplies an XML verification tool called ILINT. This chapter covers the following topics:

- [What is ILINT?](#)
- [Before Using ILINT](#)
- [Using ILINT](#)

### 3.1 What is ILINT?

Integrator LINT (ILINT) is a tool that helps you validate target metadata while defining new target types: You can perform real-time validation for the XML you write for target, instance, and collection metadata while you are writing the target metadata. ILINT also allows you to test runtime data collection for a new target type to validate data correctness.

This tool can also be used as an integrator certification tool to ensure that target type metadata conforms to acceptable standards.

#### 3.1.1 What types of validation does ILINT perform?

ILINT allows you to perform two types of validation: static, which checks the correctness of your XML and dynamic, which checks the correctness of the data collected. ILINT validates for proper XML syntax by comparing target type metadata files against their respective DTD's (\$AGENT\_HOME/sysman/admin/dtds) and management agent semantics.

##### 3.1.1.1 Static XML Validation

ILINT performs a static XML validation by performing heuristic checks, checking Management Repository constraints, and validating XML syntax on the following files:

- <target\_metadata>.xml  
Directory Location: \$AGENT\_HOME/sysman/admin/metadata
- <target\_collection>.xml  
Directory Location: \$AGENT\_HOME/sysman/admin/default\_collection
- targets.xml (target list file)

Directory Location: \$AGENT\_HOME/sysman/emd

---

---

**Note:** In general, the targets.xml file should not be edited manually. However, when developing new metadata files, you may need to add specific target instance information to the target instance file for testing purposes. The targets.xml file must be structurally and syntactically correct because the Management Agent relies on the instance-specific information contained within the file.

---

---

Specifically, ILINT validates the metadata and, if it exists, the default collection for a specific target type using a validating parser. ILINT uses all target properties (both static and dynamic) as defined in the targets.xml file.

#### **Heuristic Constraint Checks'**

- Are too many metrics being collected? If more than 50 metrics are collected for a given target type, ILINT generates a warning message.
- Are too many targets being monitored? If more than 50 targets are being monitored for a given target type, ILINT generates a warning message.
- Are metric collections intervals too short? If a specific metric collection interval is less than five minutes, ILINT generates a warning message.

#### **Target Metadata Limitations Based on Management Repository Constraints**

- Verify that column name widths do not exceed 64 characters.
- Verify that no more than 5 key columns have been set.
- Check for legal characters in metadata name, type, metric name, and column names.

#### **3.1.1.2 Dynamic Validation**

Dynamic validation involves running specific metrics through the fetchlets to validate correctness and view output.

Dynamic validation allows you to check the correctness of the data collected for a target type. ILINT performs dynamic XML validation by executing all metrics defined for the new target and displaying viewable output in much the same way as the metric browser. For more information about the metric browser, see "[Validate your new target type definitions.](#)" on page 2-10.

While performing dynamic validation, ILINT executes the specified metric and checks for errors using all properties (dynamic and static). ILINT then generates an output table containing:

- Target list
- Metric list for each target
- Metric results

#### **3.1.1.3 Metadata Version Checking**

In order for the Enterprise Manager Grid Control console to display metrics for a specific target type, syntactic rules must be followed. ILINT can also be used to verify that the current version of a metadata file is compatible with an earlier version of the same metadata file. This compatibility check is stringent: Two target metadata files being compared by ILINT must be syntactically identical.

## 3.1.2 ILINT Output

ILINT generates two types of messages, each directed to different output file streams: STDOUT and STDERR.

### Message Types

- **Informational/Warning**--Generated when the XML is technically accurate, but may cause data collection/interpretation inaccuracies or framework performance issues. Tabular data generated during dynamic validation is also part of this message class.

Informational and Warning messages are printed to STDOUT.

- **Error**--The XML is technically inaccurate or parameters specified violate management repository constraints.

Error messages are printed to STDERR.

### Specifying Separate Output Files

You can specify separate STDOUT and STDERR output files when running ILINT. File specification may differ depending on the operating system and/or shell used for your environment.

#### *Example 3-1 Specifying STDOUT and STDERR (ksh, bash, and sh)*

```
emctl ilint ... 1>my_stdout_file 2>my_stderr_file
```

#### *Example 3-2 Specifying STDOUT and STDERR (csh and tcsh)*

```
(emctl ilint ... >my_stdout_file) >&my_stderr_file
```

### Generating Syntactically Formatted XML Output

ILINT optionally allows you to generate a syntactically consistent view of a metadata file. For more information on ILINT options and generating valid XML, see "[Generating Syntactically Correct XML](#)" on page 3-7.

## 3.2 Before Using ILINT

In order to validate an XML file, ILINT needs to know the location of the DTD (Document Type Declaration) associated with the XML file being validated. Make sure that the DOCTYPE directive specifying the relative path to the appropriate DTD is included in each XML file.

The following DTDs are used within the Enterprise Manager framework:

- \$AGENT\_HOME/sysman/admin/dtds/TargetMetadata.dtd
- \$AGENT\_HOME/sysman/admin/dtds/TargetCollection.dtd
- \$AGENT\_HOME/sysman/admin/dtds/TargetInstance.dtd

The following example shows a DOCTYPE declaration that should be used when testing a target collection XML file located in the \$AGENT\_HOME/sysman/admin/default\_collection directory or an instance specific collection file in the \$AGENT\_HOME/sysman/admin/collection directory.

#### *Example 3-3 DOCTYPE Declaration*

```
<!DOCTYPE TargetCollection SYSTEM "../dtds/TargetCollection.dtd">
```

Private external DTDs (DTDs shared between multiple documents and intended for use by a single author or group of authors) are identified by the keyword SYSTEM. In this case, anyone having the appropriate OS privileges to the machine running the Management Agent is considered an author.

### 3.3 Using ILINT

ILINT is part of the Enterprise Manager command line utility and is accessed as an `emctl` option. ILINT uses the following syntax:

```
emctl ilint [-o <test_name> [-p]] -m <target_metadata.xml> [-m2 <compare_
metadata.xml>] [-c <target_collection.xml>] [-i <target_instance.xml>] [-t <target_
name>] [-r <metric_name>] [-d <0|1|2|3>]
```

**Table 3–1 ILINT Options**

Option	Description
<code>-o &lt;test_name&gt;</code>	Automatically locates the metadata directory, default collection directory and collection file, and the <code>targets.xml</code> file.  When the "-o" option is specified, the following applies: <ul style="list-style-type: none"> <li>Just a file name rather than a full path and file name should be used when specifying the value for any of the options: <code>-m</code>, <code>-m2</code>, <code>-c</code>, <code>-i</code></li> <li>Normally, option <code>-c</code> (target collection file) should be omitted.</li> <li>Normally, option <code>-i</code> (target instance file) should be omitted.</li> <li>If <code>-p</code> is specified, the formatted metadata output file(s) have the <code>test_name</code> as their prefix.</li> </ul>
<code>-p</code>	Generate syntactically formatted XML for the metadata file. This option can only be used if the <code>-o</code> option is specified.
<code>-m &lt;target_metadata.xml&gt;</code>	Full path and file name of the metadata file to be checked. Note: If the <code>-o</code> option is specified, the value must be just a file name--the full path must be omitted. <b>This is the only required option to ILINT.</b>
<code>-m2 &lt;compare_metadata.xml&gt;</code>	Full path and file name of a second metadata file to be compared against the target metadata file specified by the <code>-m</code> option. This option can be used to check version compatibility between metadata files. Note: If the <code>-o</code> option is specified, the value must be just a file name--the full path must be omitted.
<code>-c &lt;target_collection.xml&gt;</code>	Full path and file name of the target collection file. Note: If the <code>-o</code> option is specified, the value must be just a file name--the full path must be omitted.
<code>-i &lt;target_instance.xml&gt;</code>	Full path and file name of the target instance file. Note: If the <code>-o</code> option is specified, the value must be just a file name--the full path must be omitted.
<code>-t &lt;target_name&gt;</code>	Name of the target (as specified in the <code>targets.xml</code> file) to be validated.
<code>-r &lt;metric_name&gt;</code>	Name of the metric to be executed.



**Table 3–1 (Cont.) ILINT Options**

Option	Description
-d (0-3)	Sets the debug level (default value is 1). This option specifies the type of information supplied by ILINT. <ul style="list-style-type: none"> <li>■ 0=Error messages only.</li> <li>■ 1=Warning and Error messages.</li> <li>■ 2=Informational, Warning, and Error messages.</li> <li>■ 3=All messages: Debug, Informational, Warning, and Error messages.</li> </ul>

### 3.3.1 ILINT Examples

The following examples demonstrate common ILINT usage scenarios discussed previously.

#### 3.3.1.1 Static Validation

##### Usage

```
emctl ilint
  -o <test_name>
  -d <0-3>
  -i targets.xml
  -m <target_metadata.xml>
  -t <target_name>
```

##### **Example 3–4 Validating a Target Metadata File**

```
> emctl ilint -o meta_test -d 0 -i targets.xml -m my_database.xml -t my_database3
```

The metadata directory and file (-m my\_database.xml) is located automatically and associated with the test name "meta\_test" (-o meta\_test). The metadata file is validated against the database "my\_database3" (-t my\_database3) which is defined in the targets.xml file. Only error messages are returned (-d 0).

##### **Example 3–5 Validating a Target Collection File**

```
> emctl ilint -o collect_test -d 3 -i targets.xml -m co_database.xml -c my_
collection.xml -t my_database3
```

The collection directory and file (-c my\_collection.xml) is located automatically and associated with the test name "collect\_test" (-o collect\_test). Because this validation is run in full debug mode (-d 3), all error, warning, and informational messages are returned.

##### **Example 3–6 Limited Metadata Validation (target type metadata file validation only)**

```
emctl ilint -o test_name -d 0 -i non_existent_targets.xml -d 0 -m my_target_
metadata.xml -t non_existent_target
```

As shown in this example, you can perform a limited target type metadata file XML validation by specifying a non-existent targets.xml file and a non-existent target. Limited validation quickly checks XML syntax used in the target metadata file as well as any limitations in metadata due to repository constraints.

### 3.3.1.2 Dynamic Validation

#### Usage

```
emctl ilint
  -o <test_name>
  -d <0-3>
  -i targets.xml
  -m <target_metadata.xml>
  -t <target_name>
  -r <metric_name>
```

#### **Example 3-7 Validating a Metric**

```
emctl ilint -o test_dynamic -d 0 -i targets.xml -m database.xml -t payroll_db -r
Database_Resource_Usage
```

In this example, ILINT validates the database metric "Database\_Resource\_Usage." The metadata directory and file (-m database.xml) is located automatically and associated with the test name "test\_dynamic" (-o test\_dynamic). This metric is validated against the target database named "payroll\_db" (-t payroll\_db) which is defined in the targets.xml file. Only error messages are returned (-d 0).

### 3.3.1.3 Checking Metadata Version Compatibility

#### Usage

```
emctl ilint
  -o <test_name>
  -d <0-3>
  -i targets.xml
  -m <current_target_metadata.xml>
  -m2 <old_target_metadata.xml>
  -t <target_name>
```

#### **Example 3-8 Validating Target Metadata Versions**

```
emctl ilint -o version_test -d 3 -i targets.xml -m database.xml -m2 database.xml
-t payroll_db
```

In this example, ILINT compares two versions of the target type metadata file "database.xml." Because the -o option is used, ILINT looks for the most recent version of "database.xml" in \$AGENT\_HOME/sysman/admin/metadata and the older version of this file in \$AGENT\_HOME/sysman/admin/metadata/old.

---

---

**Note:** If you do not want to use the default directory locations, omit the -o <test\_name> option and specify absolute pathnames for -m and -m2.

---

---

### 3.3.1.4 Generating Syntactically Correct XML

#### Usage

```
emctl ilint
  -o <test_name>
  -i targets.xml
  -m <target_metadata.xml>
  -m2 <old_target_metadata.xml> (optional)
  -p
```

XML output is sent to a file named <test\_name>.xml.tmp1

#### **Example 3–9** Generating Syntactically Correct XML

```
emctl ilint -o test_generate -i targets.xml -m my_database.xml -p
```

The XML output from this example is sent automatically to the file "test\_generate.xml.tmp1." If the "-m2" option is specified, the XML output for the second target metadata file is sent to "test\_generate.xml.tmp2"

### 3.3.2 Usage Notes

- If the XML/DTD cannot be parsed/validated the error messages tend to be less intuitive. The Java XML parser generates more informative messages.
- The DTD path in the DOCTYPE directive in the XML file must be relative.
- In debug mode, the full information about a metric whose collection depends on the value of a ValidIf element is not displayed.



---

---

## Adding Reports

Defining new target types in Enterprise Manager via Management Plug-ins also provides you with the opportunity to add new report definitions. Plug-ins allow you to add permanent (SYSTEM) target type-specific report definitions to Enterprise Manager using the Information Publisher PL/SQL API.

This chapter covers the following:

- [What You Get](#)
- [Report Definition File](#)
- [Creating a Report Definition File](#)
- [PL/SQL Application Programmer Interface](#)
- [Development Guidelines](#)

---

---

**Note:** You must have a working knowledge of SQL and PL/SQL before using this API. Refer to the *Oracle Database PL/SQL User's Guide and Reference* for more information.

---

---

### 4.1 What You Get

Adding report definitions via Management Plug-in creates target type-specific SYSTEM reports. SYSTEM report definitions are handled differently than definitions created through the Information Publisher user interface. SYSTEM reports are permanent and cannot be deleted or edited by Enterprise Manager administrators. You can add multiple report definitions to a Management Plug-in, thus allowing you to associate multiple reports with a specific target type.

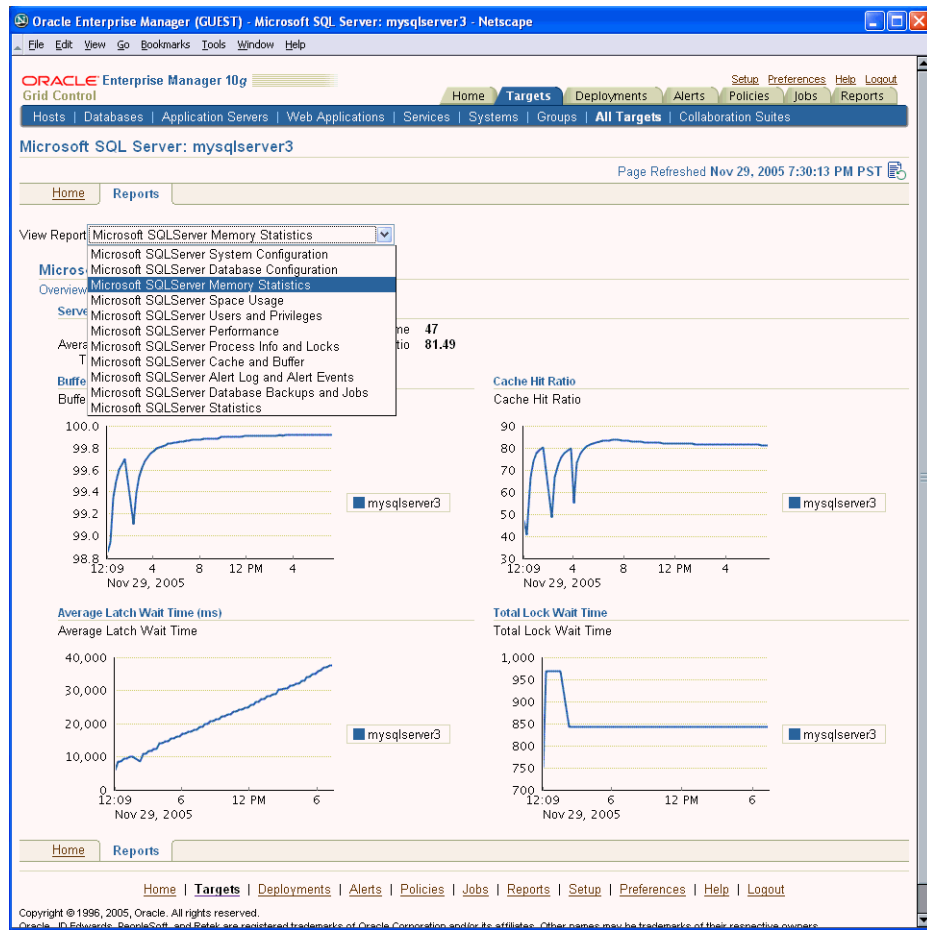
Adding SYSTEM report definitions using Management Plug-ins and the Information Publisher API allows users to access reports from two areas of the Enterprise Manager console:

- Reports page of the target home page (optional).
- Information Publisher's Report Definition page.

#### 4.1.1 Reports Page (Target Home Page)

In the report definition file, you can specify whether or not a report is available from the Reports page of the target home page. Report definitions you add to the Reports page are available from the View Reports drop-down menu list. The following figure shows the Reports page for a Microsoft SQL Server target.

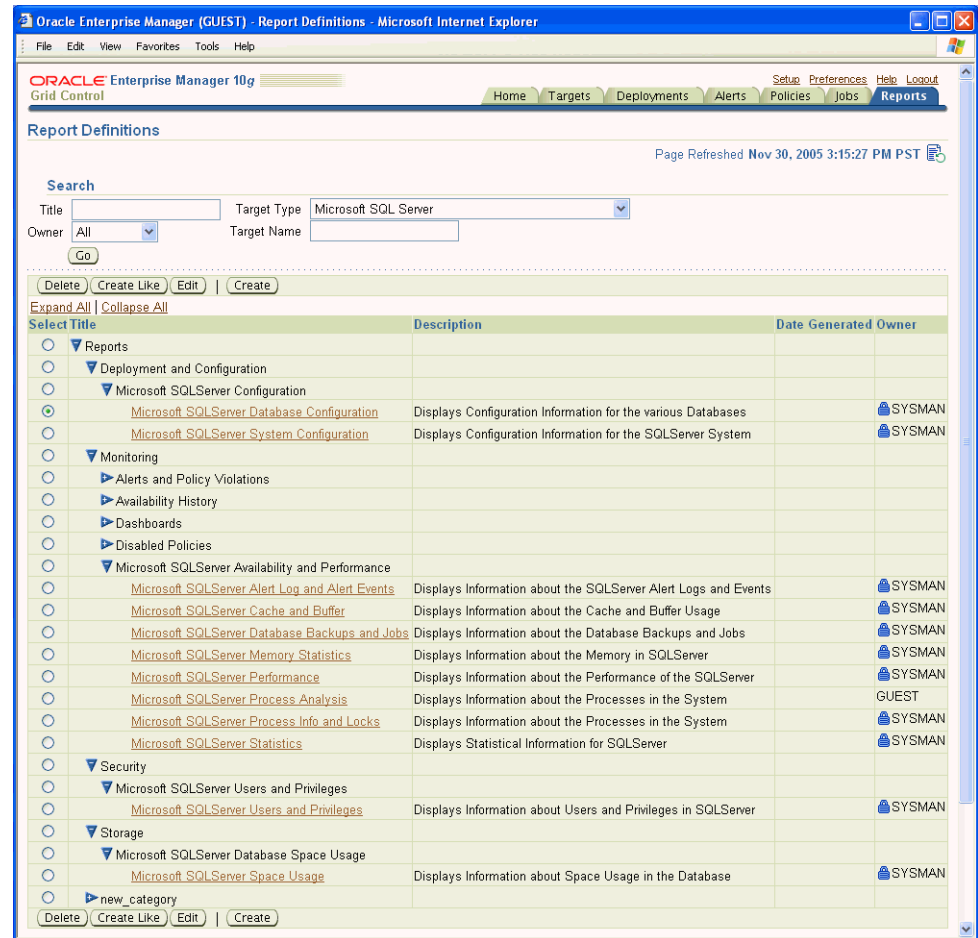
**Figure 4–1 Microsoft SQL Server Reports Page**



### 4.1.2 Report Definitions Page

Regardless of whether you specify that a report appear in the Reports page, all report definitions added via Management Plug-ins are available from Information Publisher’s Report Definitions page. As with out-of-box SYSTEM report definitions, those added via Management Plug-ins are organized according to report category and subcategory. SYSTEM report definitions cannot be deleted from the Enterprise Manager console. [Figure 4–2, "Report Definitions Page"](#) shows available report definitions for Microsoft SQL Server target types.

Figure 4–2 Report Definitions Page

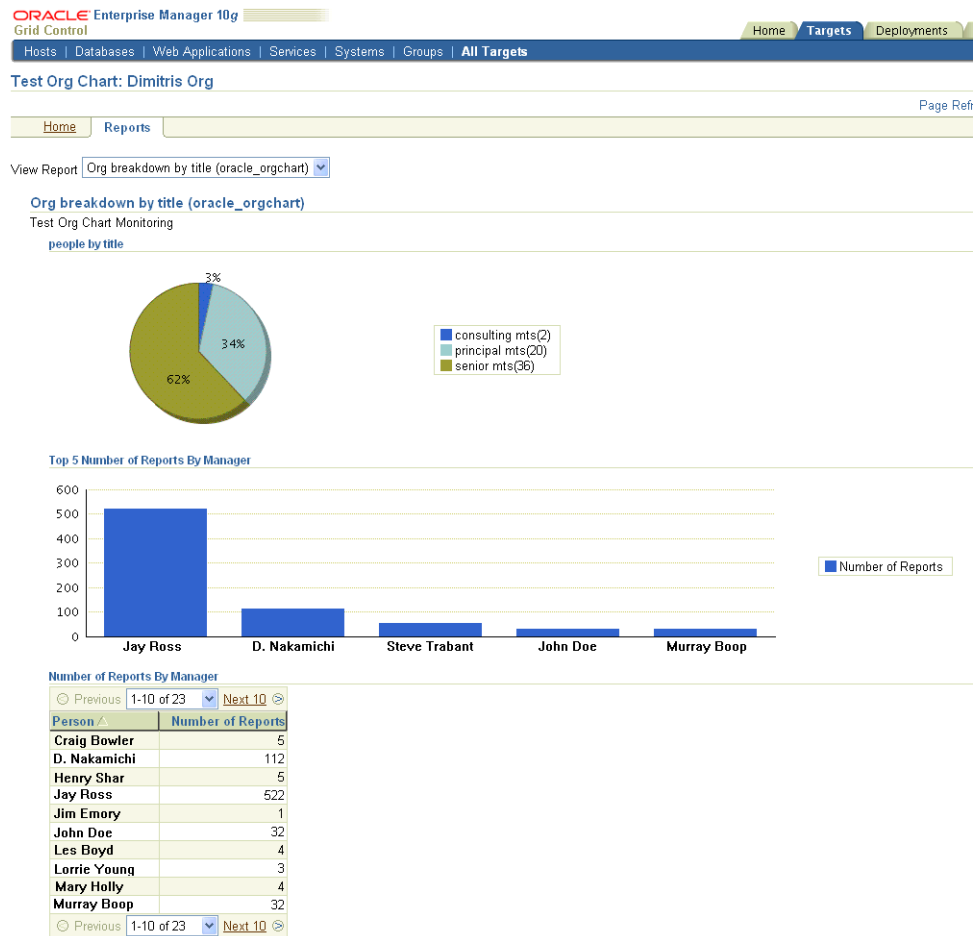


## 4.2 Report Definition File

A report definition file is structured as a conventional PL/SQL block and contains code to extract pertinent information from the Management Repository and the report elements used to format and display that data. The Information Publisher PL/SQL API allows you specify the report elements and parameters that you normally specify interactively when creating a report definition from the Enterprise Manger console.

The following example shows the Reports page for an "oracle\_orgchart" target type that was added as a Management Plug-in. A single report titled "Org breakdown by title (oracle\_orgchart)" is shown in Figure 4–3. This report consists of three major areas: a pie chart showing the percentage distribution of people by title, a bar chart showing the top five managers having the most people reporting to them, and a table listing all managers in the organization the total number of direct reports.

**Figure 4–3 Oracle Organization Report**



The report definition used for this example utilizes two report element types: *Chart from SQL* (pie and bar), and *Table from SQL*. In general, these two report elements will meet most reporting needs.

## 4.3 Creating a Report Definition File

Management Plug-ins allow you to define as many report definition files as is required for a particular target type. The content of a report definition file consists of a PL/SQL block that uses the Information Publisher PL/SQL API (part of the MGMT\_IP package) to construct a report.

### 4.3.1 Report Definition File Development Process

The process of developing a valid report definition file involves three steps:

1. Define SQL or PL/SQL queries used to extract information from the Management Repository.
2. Create a test report interactively from the Enterprise Manager console.
3. Use the PL/SQL API to create a report definition file.



### Define SQL or PL/SQL queries used to extract information from the Management Repository.

The first step in creating your report definition is to create the SQL or PL/SQL queries used to extract the requisite report information from the Management Repository. Enterprise Manager provides management views with which you can safely extract data from the Management Repository without reading from the base tables. Using repository views protects your queries from changes to the repository schema that may occur in future releases and ensures your SYSTEM report definitions remain functional. A complete listing of repository views can be found in [Chapter 6, "Management Repository Views"](#) on page 6-1.

The following query was used to extract repository information about the distribution of employee classifications within an Oracle organization. The query uses the MGMT\$METRIC\_CURRENT repository view.

```
select 'senior mts', count(value) from mgmt$metric_current
where metric_column = 'Title' and LOWER(value) like '%senior member%' and
      target_guid = ??EMIP_BIND_TARGET_GUID??
union
select 'principal mts', count(value) from mgmt$metric_current
where metric_column = 'Title' and LOWER(value) like '%principal member%' and
      target_guid = ??EMIP_BIND_TARGET_GUID??
union
select 'consulting mts', count(value) from mgmt$metric_current
where metric_column = 'Title' and LOWER(value) like '%consulting%' and
      target_guid = ??EMIP_BIND_TARGET_GUID?? ;
```

When an administrator views a report from the Enterprise Manager console that contains this SQL query string, Information Publisher automatically binds the unique identifier for the selected target to the ??EMIP\_BIND\_TARGET\_GUID?? placeholder in the SQL query string. The documentation for *Chart from SQL* and *Table from SQL* parameters provide information on this bind variable placeholder as well as others you can include in your SQL query string.

The data returned from the SQL query is then used by the *Chart from SQL* report element to render the pie chart shown in the Reports page of an oracle\_orgchart target home page as shown in [Figure 4-3](#).

### Create a test report interactively from the Enterprise Manager console.

Once you have written and tested the SQL or PL/SQL query, you can use the Enterprise Manager console to generate a version of your report interactively using the *Chart from SQL* and *Table from SQL* report elements. By using the Information Publisher user-interface, you can easily prototype reports without having to create a report definition file and import Management Plug-in Archives. You can also use this method of interactive prototyping to refine your queries and ensure that the data extracted from the Management Repository and how that information is rendered in your report meets your reporting requirements. See Enterprise Manager online help for the *Chart from SQL* and *Table from SQL* 'Set Parameters' function for information and examples on how to format your queries.

### Use the PL/SQL API to create a report definition file.

Once you are satisfied with the way your report is being rendered by Information Publisher, you are ready to create the report definition file. As previously mentioned, the report definition file consists of a PL/SQL block defining the report elements and the SQL or PL/SQL queries used by the elements to extract repository information and render the report. [Example 4-1](#) shows the content of the report definition file associated with the oracle\_orgchart target type.

**Example 4-1 Oracle Organization Report Definition File**

```

BEGIN

DECLARE
    l_target_types  MGMT_IP_TARGET_TYPES;
    l_param_values  MGMT_IP_PARAM_VALUE_LIST;
    l_target_type   MGMT_IP_TARGET_TYPES;
    l_report_guid   RAW(16);
    l_element_guid  RAW(16);
    l_report_title_nlsid  VARCHAR2(128);
    l_report_owner  VARCHAR(100);
    l_curr_order   NUMBER;
    l_curr_row     NUMBER;

BEGIN

-- specify the target type associated with this report
-- in this case the target type is 'oracle_orgchart'
l_target_type := MGMT_IP_TARGET_TYPES();
l_target_type.extend(1);
l_target_type(1) := 'oracle_orgchart';
l_report_title_nlsid := 'Org breakdown by title (oracle_orgchart)<>';
l_report_owner := mgmt_user.get_repository_owner;

-- create a report definition for the report
l_report_guid := mgmt_ip.create_report_definition (
    p_title_nlsid           => l_report_title_nlsid,
    p_description_nlsid    => 'Charts showing the breakdown by title',
    p_owner                 => l_report_owner,
    p_category_nlsid       => 'Test Org Chart Reports',
    p_sub_category_nlsid   => 'Interesting Org Data',
    p_late_binding_target_types => l_target_type,
    p_show_table_of_contents => 0,
    p_system_report        => 1,
    p_component_name       => l_target_type(1)
);

-- Add report so that it shows up under "reports" tab in default home page
mgmt_mp_homepage.add_report (
    p_target_type           => l_target_type(1),
    p_report_title         => l_report_title_nlsid,
    p_report_owner         => l_report_owner,
    p_report_order         => 1
);

-- create the first element in the report
-- it will be a text element with a description of the report contents

-- set the parameters for the styled text element
-- we'll provide a message and a display style
l_param_values := MGMT_IP_PARAM_VALUE_LIST();
l_param_values.extend(2);
l_param_values(1) := MGMT_IP_PARAM_VALUE_RECORD(
    'oracle.sysman.eml.ip.render.elem.TextParamBean.textMessage',
    'Test Org Chart Monitoring');
l_param_values(2) := MGMT_IP_PARAM_VALUE_RECORD(
    'oracle.sysman.eml.ip.render.elem.TextParamBean.textStyle',
    'OraInstructionText');

-- initialize the sequencing variables

```

```

-- l_curr_order should sequentially increase from 1 to the number of elements in
-- the report
-- l_curr_row indicates the row on which to display an element
  l_curr_order := 1;
  l_curr_row := 1;

  l_element_guid := mgmt_ip.add_element_to_report_def (
    p_report_guid      => l_report_guid,
    p_element_name_nlsid => 'IPMSG_STYLED_TEXT',
    p_element_type_nlsid => 'IPMSG_ANY_TARGET_TYPE',

    p_header_nlslid    => null,
    p_element_order    => l_curr_order,
    p_element_row      => l_curr_row,
    p_parameters       => l_param_values,
    p_targets          => null

  );

-- the second element in the report is a pie chart showing
-- the count of employees by title

  l_param_values := mgmt_ip_param_value_list();
  l_param_values.extend(3);
  l_param_values(1) := mgmt_ip_param_value_record(
    'oracle.sysman.eml.ip.render.elem.sqlStatement',
    'select ''senior mts'', count(value)
      from mgmt$metric_current where metric_column = ''Title'' and LOWER(value)
      like ''%senior member%'' and target_guid = ??EMIP_BIND_TARGET_GUID??
union
  select ''principal mts'', count(value)
      from mgmt$metric_current
  where metric_column = ''Title'' and LOWER(value)
  like ''%principal member%'' and target_guid = ??EMIP_BIND_TARGET_GUID??
union
  select ''consulting mts'', count(value)
      from mgmt$metric_current where metric_column = ''Title'' and
      LOWER(value) like ''%consulting%'' and
      target_guid = ??EMIP_BIND_TARGET_GUID?? '
  );
  l_param_values(2) := mgmt_ip_param_value_record(
    'oracle.sysman.eml.ip.render.elem.ChartParamController.chartType',
    'pieChart');
  l_param_values(3) := mgmt_ip_param_value_record(
    'oracle.sysman.eml.ip.render.elem.ChartParamController.pieShowSlicePercentLabels',
    'true');

  l_curr_order := l_curr_order + 1;
  l_curr_row := l_curr_row + 1;

-- add pie chart to report definiton
--
  l_element_guid := mgmt_ip.add_element_to_report_def (
    p_report_guid      => l_report_guid,
    p_element_name_nlsid => 'IPMSG_USER_CHART_FROM_SQL',
    p_element_type_nlsid => 'IPMSG_ANY_TARGET_TYPE',
    p_header_nlslid    => 'people by title',
    p_element_order    => l_curr_order,
    p_element_row      => l_curr_row,
    p_parameters       => l_param_values ,

```

```

        p_targets          => null

    );

--add a bar chart showing number of reports by manager
--

l_param_values := MGMT_IP_PARAM_VALUE_LIST();
l_param_values.extend(3);
l_param_values(1) := MGMT_IP_PARAM_VALUE_RECORD(
    'oracle.sysman.eml.ip.render.elem.sqlStatement',
    'select * from (select key_value, to_number(value) "Number of Reports" from
    mgmt$metric_current where target_type = ''oracle_orgchart'' and
    metric_column = ''Reports'' and value is not null and
    target_guid = ??EMIP_BIND_TARGET_GUID?? order by to_number(value) DESC )
    where rownum < 6'
);
l_param_values(2) := MGMT_IP_PARAM_VALUE_RECORD(
    'oracle.sysman.eml.ip.render.elem.ChartParamController.chartType',
    'barChart');
l_param_values(3) := MGMT_IP_PARAM_VALUE_RECORD(
    'oracle.sysman.eml.ip.render.elem.ChartParamController.width',
    '800');

l_curr_order := l_curr_order + 1;
l_curr_row := l_curr_row + 1;

l_element_guid := mgmt_ip.add_element_to_report_def (
    p_report_guid          => l_report_guid,
    p_element_name_nlsid => 'IPMSG_USER_CHART_FROM_SQL',
    p_element_type_nlsid => 'IPMSG_ANY_TARGET_TYPE',
    p_header_nlsid       => 'Top 5 Number of Reports By Manager',
    p_element_order      => l_curr_order,
    p_element_row        => l_curr_row,
    p_parameters         => l_param_values,
    p_targets            => null

);

-- the next element is a table the manager name and number of reports per manager

l_param_values := MGMT_IP_PARAM_VALUE_LIST();
l_param_values.extend(1);
l_param_values(1) := MGMT_IP_PARAM_VALUE_RECORD(
    'oracle.sysman.eml.ip.render.elem.sqlStatement',
    'select key_value "Person", to_number(value) "Number of Reports" from
    mgmt$metric_current where target_type = ''oracle_orgchart'' and
    metric_column = ''Reports'' and value is not null and
    target_guid = ??EMIP_BIND_TARGET_GUID?? order by to_number(value) DESC '
);

l_curr_order := l_curr_order + 1;
l_curr_row := l_curr_row + 1;

l_element_guid := mgmt_ip.add_element_to_report_def (
    p_report_guid          => l_report_guid,
    p_element_name_nlsid => 'IPMSG_USER_TABLE_FROM_SQL',
    p_element_type_nlsid => 'IPMSG_ANY_TARGET_TYPE',
    p_header_nlsid       => 'Number of Reports By Manager',
    p_element_order      => l_curr_order,

```

```

    p_element_row      => l_curr_row,
    p_parameters       => l_param_values ,
    p_targets          => null
  );

END;

END;
```

### 4.3.2 Report Lifecycle: Updating Report Definitions

With the ability to add report definitions to Enterprise Manager comes the responsibility of maintaining and updating the report definitions. Familiarity with the way in which Enterprise Manager handles report definitions will allow you to anticipate system behavior and plan for backwards compatibility.

When report definitions are deployed via Management Plug-in, Enterprise Manager only allows newer versions of the report definitions to be installed. Older report definitions are deleted and deregistered so as not to appear on the Reports subtab of a target home page. These actions eliminate potential version conflicts by ensuring that updated report definitions are deployed to clean systems. Enterprise Manager will not install older versions of a report definition.

Report definitions, as with Management Plug-ins in general, should be designed with backwards compatibility in mind. Future versions of report definitions should support previous versions of the target type metadata. Report definition-metadata version incompatibility will be most apparent in the following situations:

- Report definitions included with Management Plug-in version 1 AND not included with Management Plug-in version 2 will disappear when version 2 is deployed, even if version 1 deployments remain on some Management Agents..
- If version 1 and version 2 of a Management Plug-in are both deployed to the system, Management Agents will collect data based on the metadata of the version installed at that Agent; some will collect for version 1 metadata and some for version 2 metadata. Only the version 2 report definitions will be installed (appear in the Enterprise Manager console). For this reason, version 2 report definitions must support both versions of the metadata.

## 4.4 PL/SQL Application Programmer Interface

The Information Publisher PL/SQL API allows you to create a report definition file.

### 4.4.1 PL/SQL Methods for Creating Report Definitions

Use the following PL/SQL methods to create and/or manipulate report definitions when creating report definition files.

- [mgmt\\_ip.create\\_report\\_definition](#)
- [mgmt\\_ip.add\\_element\\_to\\_report\\_def](#)
- [mgmt\\_mp\\_homepage.add\\_report](#)

#### 4.4.1.1 mgmt\_ip.create\_report\_definition

Call this method to create a new report definition. Once a report definition is created, elements can be added. The `create_report_definition` method is part of the `MGMT_IP` PL/SQL package.

---



---

**Note:** All of the Management Plug-in reports should set `p_system_report` to 1. This parameter defines the report definition as a SYSTEM report, which cannot be deleted or edited by Enterprise Manager administrators. `p_owner` should be set to `mgmt_user.get_repository_owner` for all Management Plug-in reports.

The `p_component_name` must be set to the target type of the management plug-in.

---



---

### Input

<code>p_title_nlsid</code>	report title.
<code>p_description_nlsid</code>	description
<code>p_owner</code>	owner name (should be the value returned from <code>mgmt_user.get_repository_owner</code> for Plug-In reports)
<code>p_category_nlsid</code>	category name
<code>p_sub_category_nlsid</code>	subcategory name
<code>p_late_binding_target_types</code>	target type for late binding, or null if not late binding
<code>p_show_table_of_contents</code>	1=show 0=hide
<code>p_system_report</code>	1=system report, 0=end user report. This must be set to 1 for Management Plug-in reports.
<code>p_show_navigation</code>	Show navigation headers in report (tabs, etc) 1=show, 0=hide
<code>p_product_name</code>	Product name, 'EM'(default)
<code>p_component_name</code>	Product component. This must be set to the Management Plug-in target type.
<code>p_version</code>	Version, '10.2' (default)
<code>p_parameters</code>	Parameters for this report definition

### Output

Returns the GUID for this report definition.

### Code

```

FUNCTION create_report_definition (
  p_title_nlsid          IN VARCHAR2,
  p_description_nlsid   IN VARCHAR2,
  p_owner                IN VARCHAR2,
  p_category_nlsid      IN VARCHAR2,
  p_sub_category_nlsid  IN VARCHAR2,
  p_late_binding_target_types IN MGMT_IP_TARGET_TYPES DEFAULT NULL,
  p_show_table_of_contents IN NUMBER DEFAULT 0,
  p_system_report        IN NUMBER DEFAULT 1,
  p_show_navigation      IN NUMBER DEFAULT 1,
  p_product_name         IN VARCHAR2 DEFAULT 'EM',
  p_component_name       IN VARCHAR2 DEFAULT '',
  p_version              IN VARCHAR2 DEFAULT '10.2.0.1.0',
  p_parameters           IN MGMT_IP_PARAM_VALUE_LIST DEFAULT NULL
) RETURN RAW;

```

#### 4.4.1.2 mgmt\_ip.add\_element\_to\_report\_def

Call this method to add a new report element to an existing report definition. The `add_element_to_report_def` method is part of the MGMT\_IP PL/SQL package.

**Input**

p_report_guid	GUID to identify the report definition.
p_element_name_nlsid	The element name.
p_element_type_nlsid	The element type name.
p_header_nlsid	The element header or null.
p_element_order	The order of this element, 1 based.
p_element_row	The row for this element, 1 based.
p_parameters	The parameters for this element.

**Output**

Returns the GUID for this element instance.

**Code**

```
FUNCTION add_element_to_report_def(
    p_report_guid          IN RAW,
    p_element_name_nlsid  IN VARCHAR2,
    p_element_type_nlsid  IN VARCHAR2,
    p_header_nlsid        IN VARCHAR2 DEFAULT NULL,
    p_element_order       IN NUMBER,
    p_element_row         IN NUMBER,
    p_parameters           IN MGMT_IP_PARAM_VALUE_LIST,
    p_targets              IN MGMT_IP_TARGET_LIST
) RETURN RAW;
```

**4.4.1.3 mgmt\_mp\_homepage.add\_report**

Call this method to register a report for display in the Reports subtab on the target home page for a report. The add\_report method is part of the MGMT\_MP\_HOME PAGE PL/SQL package.

The input parameters p\_target\_type, p\_report\_title, and p\_report\_owner MUST be identical to the report definition being registered.

---



---

**IMPORTANT:** The value returned from mgmt\_user.get\_repository\_owner must be specified as the report owner in order for the report to appear on the Reports subtab of a target home page.

---



---

**Input**

p_target_type	The target type.
p_report_title	The report title.
p_report_owner	The report owner.
p_order	The order the report shows up in the homepage.

**Output**

None.

**Code**

```
PROCEDURE add_report(p_target_type IN VARCHAR2, p_report_title IN VARCHAR2,
    p_report_owner IN VARCHAR2, p_report_order IN NUMBER);
```

#### 4.4.1.4 mgmt\_view\_util.adjust\_tz

Call this function to convert the time zone of `v_date_in` from the `v_from_tz` time zone to the `v_to_tz` time zone.

##### Input

<code>v_date_in</code>	Date to be converted to a different time zone
<code>v_from_tz</code>	time zone of date being converted.
<code>v_to_tz</code>	time zone into which date will be converted.

##### Output

Returns the adjusted date in the new time zone.

##### Code

```
FUNCTION ADJUST_TZ(v_date_in DATE,
                  v_from_tz VARCHAR2,
                  v_to_tz VARCHAR2)
RETURN DATE;
```

## 4.4.2 PL/SQL Type Definitions

Three PL/SQL types are required to use the PL/SQL methods documented in [Section 4.4.1, "PL/SQL Methods for Creating Report Definitions"](#). The definitions for these types are shown below.

### MGMT\_IP\_TARGET\_TYPES

Use `MGMT_IP_TARGET_TYPES` type to pass the target type your report definition supports to the `create_report_definition` API as the `p_late_binding_target_types` parameter.

```
CREATE OR REPLACE TYPE MGMT_IP_TARGET_TYPES
AS TABLE OF VARCHAR(64);
```

#### *Example 4-2* MGMT\_IP\_TARGET\_TYPES

```
DECLARE
  l_target_type MGMT_IP_TARGET_TYPES;
BEGIN
  -- specify the target type associated with this report
  -- in this case the target type is 'oracle_orgchart'
  l_target_type := MGMT_IP_TARGET_TYPES();
  l_target_type.extend(1);
  l_target_type(1) := 'oracle_orgchart';
END;
```

### MGMT\_IP\_PARAM\_VALUE\_LIST and MGMT\_IP\_PARAM\_VALUE\_RECORD

Use the `MGMT_IP_PARAM_VALUE_LIST` type to pass parameter values to the `create_report_definition` API as the `p_parameters` parameter and to the `add_element_to_report_def` API as the `p_parameters` parameter.

```
CREATE OR REPLACE TYPE MGMT_IP_PARAM_VALUE_LIST
AS TABLE OF MGMT_IP_PARAM_VALUE_RECORD;
```

Use the `MGMT_IP_PARAM_VALUE_RECORD` type to create a named parameter value pair to add to an object of type `MGMT_IP_PARAM_VALUE_LIST`.



```
CREATE OR REPLACE TYPE MGMT_IP_PARAM_VALUE_RECORD
AS OBJECT (PARAM VARCHAR2(100), VALUE CLOB);
```

**Example 4–3 MGMT\_IP\_PARAM\_VALUE\_RECORD and MGMT\_IP\_PARAM\_VALUE\_LIST**

```
DECLARE
  l_param_values MGMT_IP_PARAM_VALUE_LIST;
BEGIN
  l_param_values := MGMT_IP_PARAM_VALUE_LIST();
  l_param_values.extend(2);
  l_param_values(1) := MGMT_IP_PARAM_VALUE_RECORD(
    'oracle.sysman.eml.ip.render.elem.TextParamBean.textMessage',
    'Test Org Chart Monitoring');
  l_param_values(2) := MGMT_IP_PARAM_VALUE_RECORD(
    'oracle.sysman.eml.ip.render.elem.TextParamBean.textStyle',
    'OraInstructionText');
END;
```

## 4.4.3 Element Parameters

Parameters used by some report elements dictate the operational behavior of those elements. This section lists the parameters associated with specific report elements.

### 4.4.3.1 Table Element Parameters

The Table Element is used to show a tabular view of query results. The queries must be made against management views.

- Element Name: IPMSG\_USER\_TABLE\_FROM\_SQL
- Element Type: IPMSG\_ANY\_TARGET\_TYPE

#### Time Period

Parameter Name	"oracle.sysman.eml.ip.render.elem.TimePeriodParam"
Required	No
Default Value	Null
Valid Values	"0:0" for last 24 Hours "0:1" for last 7 Days "0:2" for last 31 Days
Summary	Encoded time period.

#### Sort Column

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.initialSortColumn"
Required	No
Default Value	The first column in result set.
Valid Values	Any valid column name.
Summary	If this parameter is set, the sort column indicator will be shown for the column with this column name. If not set, the sort column indicator is shown on the first column. The SQL query should include an 'order by' clause that sorts by this column.

**Sort Order**

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.initialSortOrder"
Required	No
Default Value	"ascending"
Valid Values	"ascending" or "descending"
Summary	If this parameter is set, the sort column indicator will be shown either as ascending or descending, according to the value. If not set, the sort column indicator is shown as ascending.

**Name Value Pair Display**

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.nameValueDisplay"
Required	No
Default Value	
Valid Values	Positive integer value.
Summary	If this parameter is set and only one row is returned from the query, the results are displayed in a vertical list of name-value pairs. This value should be set to the number of name/value columns that should be displayed, normally "1".

**Number of Rows to Show**

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.numRowsToShow"
Required	No
Default Value	"10"
Valid Values	Positive integer value.
Summary	Number of rows to display at one time in the generated table. The user can scroll through additional rows using the UI controls..

**Is PL/SQL Statement**

Parameter Name	"oracle.sysman.eml.ip.render.elem.sqlStatementIsPISql"
Required	No
Default Value	"false"
Valid Values	"true" or "false"
Summary	Whether a SQL statement is PL/SQL.

**SQL or PL/SQL Statement**

Parameter Name	"oracle.sysman.eml.ip.render.elem.sqlStatement"
Required	No
Default Value	<None>
Valid Values	Any valid SQL SELECT statement.

## Summary

SQL statement can optionally bind values for targets, locale information, and start/end date. The format of the SQL statement should include a bind variable placeholders for the options to be bound.

**Bind Placeholders**

- ??EMIP\_BIND\_RESULTS\_CURSOR??  
For use with PL/SQL statement to bind a return cursor containing results for display.
- ??EMIP\_BIND\_TARGET\_GUID??  
For use with SQL or PL/SQL to bind a target GUID.
- ??EMIP\_BIND\_START\_DATE??  
For use with SQL or PL/SQL to bind a start date.
- ??EMIP\_BIND\_END\_DATE??  
For use with SQL or PL/SQL to bind an end date.
- ??EMIP\_BIND\_TIMEZONE\_REGION??  
For use with SQL or PL/SQL to bind a time zone region.
- ??EMIP\_BIND\_LOCALE\_COUNTRY??  
For use with SQL or PL/SQL to bind a locale country.
- ??EMIP\_BIND\_LOCALE\_LANGUAGE??  
For use with SQL or PL/SQL to bind a locale language.

There should be no semi-colon (;) appended to the end of the SQL statement unless it is a PL/SQL statement.

**Example 4–4 Specifying an Anonymous PL/SQL Block as a Parameter to an Element Definition**

```
l_param_values(1) := MGMT_IP_PARAM_VALUE_RECORD(
  'oracle.sysman.eml.ip.render.elem.sqlStatementIsPlSql',
  'true');
l_param_values(2) := MGMT_IP_PARAM_VALUE_RECORD(
  'oracle.sysman.eml.ip.render.elem.sqlStatement',
  'BEGIN
    DECLARE
      rep_tz          VARCHAR2(200);
      user_tz_in     VARCHAR2(200);
      TYPE CURSOR_TYPE IS REF CURSOR;
      result_cursor_out CURSOR_TYPE;
      start_date_in DATE DEFAULT NULL;
      end_date_in   DATE DEFAULT NULL;
      query_string   VARCHAR(6000);
    BEGIN
      result_cursor_out := ??EMIP_BIND_RESULTS_CURSOR??;
      start_date_in := ??EMIP_BIND_START_DATE??;
      end_date_in := ??EMIP_BIND_END_DATE??;
      user_tz_in := ??EMIP_BIND_TIMEZONE_REGION??;
      select TIMEZONE_REGION into rep_tz from mgmt$target where TARGET_TYPE =
'oracle_emrep';
      query_string := 'WITH dates AS (SELECT
mgmt_view_util.adjust_tz(:1,:2,:3) as start_date,
mgmt_view_util.adjust_tz(:4,:5,:6) as end_date
from dual)
SELECT
  label,
  time,
  sum(violations) --) as violations
FROM(
```

```

(
  SELECT
    '||'NEW_LBL'||' as label,
    mgmt_view_util.adjust_tz(rollup_timestamp, :7, :8) as time,
    average as violations --new violations
  FROM
    MGMT$METRIC_HOURLY
  WHERE
    key_value in ('18', '20', '25') AND
    ROLLUP_TIMESTAMP > (SELECT start_date FROM dates) AND
    ROLLUP_TIMESTAMP < (SELECT end_date FROM dates) AND
    target_type = 'oracle_emrep' AND
    metric_name = 'TARGET_SECURITY_NEW_VIOLATIONS' AND
    metric_column = 'NEW_VIOLATIONS' )
UNION
(
  SELECT
    '||FIXED_LBL'||' as label,
    mgmt_view_util.adjust_tz(rollup_timestamp, :9, :10) as time,
    average as violations --cleared violations
  FROM
    MGMT$METRIC_HOURLY
  WHERE
    key_value in ('18', '20', '25') AND
    ROLLUP_TIMESTAMP > (SELECT start_date FROM dates) AND
    ROLLUP_TIMESTAMP < (SELECT end_date FROM dates) AND
    target_type = 'oracle_emrep' AND
    metric_name = 'TARGET_SECURITY_CLEARED_VIOLATIONS' AND
    metric_column = 'CLEARED_VIOLATIONS' )
)
)
GROUP BY time, label
ORDER BY time ASC';
OPEN result_cursor_out for query_string using start_date_in, user_tz_in, rep
_tz, end_date_in, user_tz_in, rep_tz, user_tz_in, rep_tz, user_tz_in, rep_tz;
END;
END;');

```

### Maximum Number of Rows

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.maxNumberOfRowsAllowed"
Required	No
Default Value	"2000"
Valid Values	Any scalar numeric value.
Summary	Set the maximum number of rows retrieved for display in the table. For example, show the top 10 xyz's element would set the value to "10".

### Null Data String Substitute

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.nullDataStringSubstitute"
Required	No
Default Value	""
Valid Values	A string.

Summary A string that will be substituted for null values returned.

### Split Table into Multiple Tables by Column

Parameter Name	TableRenderBean.TABLE_SPLIT_COLUMN
Parameter String	"oracle.sysman.eml.ip.render.elem.TableRender.tableSplitColumn"
Required	No
Default Value	null
Valid Values	Any valid column name.
Summary	If this parameter is set, the table will be split into separate tables with subheaders as the value in this column changes. The data should be ordered by this column.

### Column Group Header

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.columnGroupHeader"n
Required	No
Default Value	null
Valid Values	Header string to use for a column group.
Summary	This parameter provides a column header string. This column group header will span columns between the columns specified in "oracle.sysman.eml.ip.render.elem.TableRender.columnGroupStart Col"n and oracle.sysman.eml.ip.render.elem.TableRender.columnGroupEndCol"n. The n suffix is a numeric value starting with 1 for the first column group, sequentially ascending for subsequent column groups.

### Column Group Start Column

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.columnGroupStartCol"n
Required	No
Default Value	null
Valid Values	Any valid column name.
Summary	Specifies the first column for a given column group. The n suffix is a numeric value starting with 1 for the first column group, sequentially ascending for subsequent column groups.

### Column Group End Column

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.columnGroupEndCol"n
Required	No
Default Value	null
Valid Values	Any valid column name.
Summary	Specifies the last column for a given column group. The n suffix is a numeric value starting with 1 for the first column group, sequentially ascending for subsequent column groups.

### Use Separate Rows for Values within a Cell

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.useSeparateRowsColumns"
Required	No

Default Value	null
Valid Values	Comma separated list of valid column names.
Summary	If this parameter is set, the delimited values of the column with the given name specified will be displayed on separate rows within a containing row cell. More than one column can be designated for this treatment by adding comma-separated column names.

### Use Separate Rows as Delimiters

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.useSeparateRowsDelimiter"
Required	No
Default Value	, (comma)
Valid Values	Any string.
Summary	A character used to delimit tokens within a string.

### Severity Icon in Column

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.severityColumn"
Required	No
Default Value	null
Valid Values	Any valid column names.
Summary	A severity icon will be substituted for valid severity values returned. To omit an icon, your result set can contain null values in this column.

### Availability Status Icon in Column

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.availabilityStatusColumn"
Required	No
Default Value	null
Valid Values	Any valid column names.
Summary	An availability status icon will be substituted for valid values returned. To omit an icon your result set can contains null values in this column.

### Render Image in Column

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.imageFilenameColumns"
Required	No
Default Value	null
Valid Values	Comma separated list of column names.
Summary	Optional parameter to display the given image filename in the indicated columns. Indicate for which columns the given image should be rendered. Specify a comma separated list of column names. The image filename returned should contain a relative path starting with '/images' such as '/images/xyz.gif'. Normally, a SQL decode function would be used to translate a numeric value into the appropriate image filename.

### Target Type Column

Parameter Name	"oracle.sysman.eml.ip.render.elem.TableRender.targetTypeColumns"
----------------	--

Required	No
Default Value	null
Valid Values	Comma separated list of column names.
Summary	Optional parameter to indicate for which columns the value returned should be used as an internal target type to be translated into a display string for that type. Specify a comma separated list of column names.

#### 4.4.3.2 Chart Element

The Chart Element is used to show a graphical view of query results. The queries must be made against Management Repository views.

- Element Name: IPMSG\_USER\_CHART\_FROM\_SQL
- Element Type: IPMSG\_ANY\_TARGET\_TYPE

#### Chart Type

Parameter Name	"oracle.sysman.eml.ip.render.elem.ChartParamController.chartType"
Required	No
Default Value	"pieChart"
Valid Values	"barChart" or "lineChart" or "pieChart" or "timeSeriesChart" "timeSeriesBarChart"
Summary	Chart type to display.

#### Time Period

Parameter Name	"oracle.sysman.eml.ip.render.elem.TimePeriodParam"
Required	No
Default Value	Null
Valid Values	"0:0" for last 24 Hours "0:1" for last 7 Days "0:2" for last 31 Days
Summary	Encoded time period.

#### Fill

Parameter Name	"oracle.sysman.eml.ip.render.elem.ChartParamController.fill"
Required	No
Default Value	"none"
Valid Values	"none", "absolute", or "cumulative"

Summary Indicates if a line chart should fill the area under the lines.  
 "none": no fill under lines.  
 "absolute": lines are identical to "none" setting but with the area under the lines filled.  
 "cumulative": causes the values for the lines to be added or stacked, then the areas underneath the lines are filled.  
 Use caution when using the fill attribute to ensure there is no confusion for the report user as to whether the data in the chart is cumulative or absolute.

### Height

Parameter Name "oracle.sysman.eml.ip.render.elem.ChartParamController.height"  
 Required No  
 Default Value "200"  
 Valid Values *n*, where *n* is any String that will correctly parse to a positive integer.  
 Summary Sets the display height of the chart in pixels.

### Horizontal or Vertical

Parameter Name "oracle.sysman.eml.ip.render.elem.ChartParamController.visualOrientation"  
 Required No  
 Default Value "horizontal"  
 Valid Values "horizontal" or "vertical"  
 Summary Visual orientation of the chart. This attribute is only valid with the chartType attribute set to barChart or timeSeriesChart. The attribute does not affect the pieChart.

### Legend Position

Parameter Name "oracle.sysman.eml.ip.render.elem.ChartParamController.legendPosition"  
 Required No  
 Default Value "east"  
 Valid Values "default", "east", "south"  
 Summary Specifies where the legend should be placed relative to the chart.

### Is PL/SQL Statement

Parameter Name "oracle.sysman.eml.ip.render.elem.sqlStatementIsPLSql"  
 Required No  
 Default Value "false"  
 Valid Values "true" or "false"  
 Summary Set to "true" to indicate that the SQL statement is a PL/SQL statement.

### SQL or PL/SQL Statement

Parameter Name "oracle.sysman.eml.ip.render.elem.sqlStatement"  
 Required No  
 Default Value <None>



Valid Values	Any valid SQL SELECT statement or PL/SQL block.
Summary	SQL or PL/SQL statement can optionally bind values for targets, locale information, and start/end date. The format of the statement should include a bind variable placeholders for the options to be bound.
	<b>Bind Placeholders</b> <ul style="list-style-type: none"> <li>■ ??EMIP_BIND_RESULTS_CURSOR?? For use with PL/SQL statement to bind a return cursor containing results for display</li> <li>■ ??EMIP_BIND_TARGET_GUID?? For use with SQL or PL/SQL to bind a target GUID.</li> <li>■ ??EMIP_BIND_START_DATE?? For use with SQL or PL/SQL to bind a start date.</li> <li>■ ??EMIP_BIND_END_DATE?? For use with SQL or PL/SQL to bind an end date.</li> <li>■ ??EMIP_BIND_LOCALE_COUNTRY?? For use with SQL or PL/SQL to bind a locale country.</li> <li>■ ??EMIP_BIND_LOCALE_LANGUAGE?? For use with SQL or PL/SQL to bind a locale language</li> </ul> <p>There should be no semi-colon (;) appended to the end of the SQL statement unless it is a PL/SQL statement.</p>

### Stacked Bar Chart

Parameter Name	"oracle.sysman.eml.ip.render.elem.ChartParamController.stacked"
Required	No
Default Value	"false"
Valid Values	"true" or "false"
Summary	Indicates if a bar chart should be stacked.

### Chart Title

Parameter Name	"oracle.sysman.eml.ip.render.elem.ChartParamController.title"
Required	No
Default Value	<None>
Valid Values	
Summary	Chart title to identify chart for Americans with Disabilities Act compliance.

### Width

Parameter Name	"oracle.sysman.eml.ip.render.elem.ChartParamController.width"
Required	No
Default Value	"400"
Valid Values	<i>n</i> , where <i>n</i> is any String that will correctly parse to a positive integer.
Summary	Specifies the display width of the element in pixels.

### Y-Axis Label

Parameter Name	"oracle.sysman.eml.ip.render.elem.yAxisLabel"
Required	No
Default Value	<None>
Valid Values	String
Summary	If this parameter is supplied, it is used as the y-axis label for charts that have an y-axis.

### Slices as Percentage

Parameter Name	"oracle.sysman.eml.ip.render.elem.ChartParamController.pieShowSlicePercentLabels"
Required	No
Default Value	<None>
Valid Values	"true" or "false"
Summary	If this parameter is supplied, it controls whether each slice is labeled with a percentage value. This attribute is ignored for chartTypes other than pieChart.

### Show Values in Legend

Parameter Name	"oracle.sysman.eml.ip.render.elem.ChartParamController.pieValuesInLegend"
Required	No
Default Value	"value"
Valid Values	"percent", "value" or "none"
Summary	For pie charts, this parameter specifies whether values for pie slices are included in the legend along with the label for the pie slice. The default value for this attribute is "value". If specified as either "percent" or "value" then the numeric value is displayed along with the pie slice label in the form, "pie slice label (numeric value)". If "percent" is specified, then the percentage out of the total of all slice values is calculated and displayed, otherwise, the raw value of the slice is displayed. To omit a value in the legend, specify "none" as a value for this parameter. This attribute is ignored for chartTypes other than pieChart.

## 4.4.4 Metric Details Element

Classes directly involving the Metric Details report element and its parameters are located in the `oracle.sysman.eml.ip.render.elem` package. This element also accesses parameter-related constants defined in the `oracle.sysman.emSDK.eml.EmlConstants` class.

- Element Name: `IPMSG_METRIC_DETAILS`
- Element Type: `IPMSG_ANY_TARGET_TYPE`

### Target Type

Parameter Name	"oracle.sysman.eml.ip.render.elem.MetDetInternalTargetType"
Required	No
Default Value	"oracle_database"
Valid Values	Any valid internal target type name.
Summary	The type of target to be shown in the graph.

**Metric Name**

Parameter Name	"oracle.sysman.eml.ip.render.elem.MetDetSelectedMetric"
Required	Yes
Default Value	
Valid Values	Valid metric name according to target type selected.
Summary	Metric to be graphed.

**Metric Column Name**

Parameter Name	"oracle.sysman.eml.ip.render.elem.MetDetSelectedMetricColumn"
Required	Yes
Default Value	
Valid Values	Valid column name according to the metric and target type selected.
Summary	Column of metric to be graphed.

**Time Period**

Parameter Name	"oracle.sysman.eml.ip.render.elem.TimePeriodParam"
Required	No
Default Value	null
Valid Values	"0:0" for last 24 Hours "0:1" for last 7 Days "0:2" for last 31 Days
Summary	Encoded time period.

**Width**

Parameter Name	"oracle.sysman.eml.ip.render.elem.MetDetWidth"
Required	No
Default Value	300
Valid Values	<i>n</i> , where <i>n</i> is any String that will correctly parse to a positive integer.
Summary	Width of the image in pixels.

**Height**

Parameter Name	"oracle.sysman.eml.ip.render.elem.MetDetHeight"
Required	No
Default Value	300
Valid Values	<i>n</i> , where <i>n</i> is any String that will correctly parse to a positive integer.
Summary	Height of the image in pixels.

**Legend Position**

Parameter Name	"oracle.sysman.eml.ip.render.elem.MetDetLegendPosition"
Required	No

Default Value	
Valid Values	"south" (default), "east"
Summary	Position of the legend relative to the chart.

#### 4.4.5 Text Element Parameters

The Text Element is used to display any message text you wish to provide for your report.

- Element Name: IPMSG\_STYLED\_TEXT
- Element Type: IPMSG\_ANY\_TARGET\_TYPE

##### Message Text

Parameter Name	"oracle.sysman.eml.ip.render.elem.TextParamBean.textMessage"
Required	No
Default Value	"" (empty String)
Valid Values	Any message
Summary	Set the message to display in the report.

##### Message Style

Parameter Name	"oracle.sysman.eml.ip.render.elem.TextParamBean.textStyleClass"
Required	No
Default Value	"OraInstructionText"
Valid Values	"OraInstructionText" "OraTipText"
Summary	Specifies the style class for the message text to adopt when displayed.

##### Link Destination

Parameter Name	"oracle.sysmn.eml.ip.render.elem.TextParamBean.textDestination"
Required	No
Default Value	None
Valid Values	Any URI.
Summary	Specifies an optional link destination for this text element.

#### 4.4.6 Report-Wide Parameters

The following parameters apply to all reporting elements within the report definition.

##### Dynamic Time Selector

You can provide a dynamic time period selector for your report definition that allows the report user to choose a specific time period with which to view the report. The dynamic time period option is available only when viewing reports from the Reports tab, is not available on a target home page report. The time period used on a target home page report is the default time period set for the report definition. To specify this using the PL/SQL API when creating your report definition file, set the following parameters for your report definition.

**Example 4–5 Dynamic Time Selector Report Definition Parameters**

```

l_param_values := MGMT_IP_PARAM_VALUE_LIST();
l_param_values.extend(3);

-- this report has a time period associated with it
l_param_values(1) := MGMT_IP_PARAM_VALUE_RECORD(
  'oracle.sysman.eml.ip.render.elem.TimePeriodOption',
  'true');

-- the user can customize the time period while viewing the report
l_param_values(2) := MGMT_IP_PARAM_VALUE_RECORD(
  'oracle.sysman.eml.ip.render.elem.TimePeriodUserCust',
  'true');

-- set the default value to last 24 hours
l_param_values(3) := MGMT_IP_PARAM_VALUE_RECORD(
  'oracle.sysman.eml.ip.render.elem.TimePeriodParam',
  '0:0');

```

Pass `l_param` values to `mgmt_ip.create_report_definition` as the `p_parameters` argument.

If you are using *Table from SQL* or *Chart from SQL* report elements, you can structure your SQL statement such that the start and end dates will be bound automatically for you by Information Publisher. You achieve this by inserting placeholders (for example, `??EMIP_BIND_START_DATE??`) for the start and end date values as shown in [Example 4–6](#)

**Example 4–6 Automatic Binding of Start and End Dates**

```

'SELECT COLUMN_LABEL, ROLLUP_TIMESTAMP, AVERAGE
  FROM MGMT$METRIC_HOURLY
 WHERE TARGET_GUID = ??EMIP_BIND_TARGET_GUID??
 AND METRIC_LABEL = 'Load'
 AND KEY_VALUE = ' '
 AND ROLLUP_TIMESTAMP > ??EMIP_BIND_START_DATE??
 AND ROLLUP_TIMESTAMP < ??EMIP_BIND_END_DATE??
 ORDER BY ROLLUP_TIMESTAMP'

```

See the online help documentation for *Table from SQL* or *Chart from SQL* for detailed information.

## 4.5 Development Guidelines

Oracle recommends adhering to the following guidelines when defining the PL/SQL for a report definition file:

- **Use the PL/SQL API to create your report definition.**

Report definitions should be created using the PL/SQL API rather than non-standard coding constructs. Using the PL/SQL API insulates your code from any Management Repository schema changes.

- **Every report definition added to a Management Plug-in must be defined as a SYSTEM report.**

Each call to the `create_report_definition` method must pass `p_system_report => 1`. SYSTEM report definitions are handled differently in the Enterprise Manager console. For example, SYSTEM report definitions cannot be deleted or edited by administrators.

- **Specifying the report owner.**

You must specify the value returned from `mgmt_user.get_repository_owner` as the owner for each report definition. Report definitions specifying any owner other than `mgmt_user.get_repository_owner` will not appear in the Reports subtab of the target home page. These report definitions will, however, appear in the Information Publisher Report Definitions page.

- **The component name must be set to the target type.**

The component name must be set to the target type in order for Enterprise Manager to associate specific report definitions with a particular Management Plug-in. For example,

```
BEGIN

-- specify the target type associated with this report
-- in this case the target type is 'oracle_orgchart'
l_target_type := MGMT_IP_TARGET_TYPES();
l_target_type.extend(1);
l_target_type(1) := 'oracle_orgchart';
l_report_title_nlsid := 'Org breakdown by title (oracle_orgchart)<>';
l_report_owner := mgmt_user.get_repository_owner;

-- create a report definition for the report
l_report_guid := mgmt_ip.create_report_definition (
  p_title_nlsid           => l_report_title_nlsid,
  p_description_nlsid    => 'Charts showing the breakdown by title',
  p_owner                => l_report_owner,
  p_category_nlsid      => 'Test Org Chart Reports',
  p_sub_category_nlsid  => 'Interesting Org Data',
  p_late_binding_target_types => l_target_type,
  p_show_table_of_contents => 0,
  p_system_report       => 1,
  p_component_name      => l_target_type(1)
);
```

### When Using *Chart from SQL* and *Table from SQL* elements

- If your element accepts a single non-aggregate target (only), which is the case for most Management Plug-in target types, you can take advantage of automatic time zone date adjustment built into the *Chart from SQL* and *Table from SQL* elements by setting the `'oracle.sysman.eml.ip.render.elem.adjustTimes'` parameter on your element to `'true'`. When this parameter is set, the start and end dates bound to your SQL query will be adjusted from the report time zone to the target time zone. Conversely, dates returned from the query will be adjusted from the target time zone to the report time zone.
- If your element accepts multiple targets or aggregate targets, you are responsible for handling time zone adjustment for your date values. You can obtain the report time zone from the `??EMIP_BIND_TIMEZONE_REGION??` bind variable. In order for the report viewer to understand the dates shown, dates displayed in a report must either conform to the report time zone or explicitly display the time zone associated with each date. The following examples illustrate common use cases.

**Example 4–7 Adjusting a date returned in your select statement from the time zone of a given target to the report time zone.**

```
select mgmt_view_util.adjust_tz(tbl.date, tgt.timezone_region,
??EMIP_BIND_TIMEZONE_REGION??)
from mgmt$target tgt, sometable tbl
```

where <your where clause here>

**Example 4–8 Adjusting a report time period start and end dates used in the WHERE clause of your SELECT statement from the report time zone to your targets time zone**

```
select <your selected columns here>
from mgmt$target tgt, sometable tbl
where
    tgt.target_guid = ??EMIP_BIND_TARGET_GUID?? and
    tbl.Mydate > MGMT_VIEW_UTIL.ADJUST_TZ(
??EMIP_BIND_START_DATE??,
??EMIP_BIND_TIMEZONE_REGION??,
tgt.TIMEZONE_REGION)
and
    tbl.Mydate < MGMT_VIEW_UTIL.ADJUST_TZ(
??EMIP_BIND_END_DATE??,
??EMIP_BIND_TIMEZONE_REGION??,
tgt.TIMEZONE_REGION)
```

**Recommended Coding Practice**

When calling the PL/SQL API methods, you should use named notation rather than positional notation. If you have => in your call, you are using named notation. For example,

```
l_report_guid := mgmt_ip.create_report_definition (
    p_title_nlsid           => 'Org breakdown by title (oracle_orgchart)',
    p_description_nlsid     => 'Charts showing the breakdown by title',
    p_owner                 => mgmt_user.get_repository_owner,
    p_category_nlsid       => 'Test Org Chart Reports',
    p_sub_category_nlsid   => 'Interesting Org Data',
    p_late_binding_target_types => l_target_type,
    p_late_binding_multi_targets => 0,
    p_show_table_of_contents => 0,
    p_system_report        => 1,
    p_component_name       => 'oracle_orgchart'
);
```

Using named notation insulates your report definition from any code changes to future releases of the PL/SQL API and also helps make your code self-documenting.





---

---

## Management Using Web Services

Web Services are loosely coupled software components that expose an external interface via Web Service Definition Language (WSDL). These components communicate across a network using a standard messaging protocol called Simple Object Access Protocol (SOAP). SOAP communication is supported by the Management Agent's SOAP fetchlet.

---

---

**Note:** For more information about the Web Services standard, see the World Wide Web Consortium (W3C) website:

[HTTP://www.w3.org](http://www.w3.org)

---

---

Using Enterprise Manager's Management Plug-in mechanism, you can add a new target type that exposes a WSDL management interface and monitor new target instances of this type using SOAP messages. Once a target instance of this type (one that exposes a Web Service management interface) has been added to the Enterprise Manager framework, Enterprise Manager functionality, such as notifications, jobs, and reporting is automatically extended to it.

This chapter covers the following:

- [Monitoring Using Web Services in Enterprise Manager](#)
- [Generating Target Metadata and Collection Files](#)
- [Creating a Management Plug-in Archive](#)
- [Importing the Management Plug-in](#)
- [Deploying the Management Plug-in to the Management Agent](#)
- [Adding a Target Instance](#)
- [Viewing Monitored Metrics](#)

### 5.1 Monitoring Using Web Services in Enterprise Manager

Using Enterprise Manager to monitor targets that expose a Web Services management interface entails defining a new target type via Management Plug-ins.

---

---

**Note:** This chapter assumes knowledge of Management Plug-ins and the requisite target definition files. For information on developing and deploying Management Plug-ins, see [Chapter 1, "Extending Monitoring"](#) and [Chapter 2, "Developing a Management Plug-in"](#).

---

---

Setting up Enterprise Manager to monitor a target via its Web Service management interface involves the following steps:

1. Generate the target metadata and default collection files to be added to the Management Plug-in.
2. Create a Management Plug-in Archive containing the Management Plug-in(s) defining a Web services target type. An archive may contain more than one Management Plug-in.
3. Import the Management Plug-in into Enterprise Manager.
4. Deploy the Management Plug-in to the appropriate Management Agent(s).

## 5.2 Generating Target Metadata and Collection Files

The first step is to create the requisite target definition files for your Web Service: Target metadata and default collection. At the very least, these two files are required to create a Management Plug-in. You generate the target metadata and default collection file using Web Services command line tool.

### 5.2.1 Web Services Command Line Tool

The command line tool simplifies creating new Management Plug-ins by generating the requisite target metadata and collection files, both of which are required to collect metrics from resources that support the WSDL interface. Information retrieval is achieved via the SOAP fetchlet that is integrated with the Management Agent.

This tool parses a specified WSDL file for all the operations and allows you to pick one or more operations to be invoked. If there are multiple port types specified in the WSDL file, the tool will prompt you to select one of them. Operations are listed along with their parameters. An operation in a Web Service can be one of four types:

- One Way
- Request Response
- Solicit Response
- Notification

Of the aforementioned types, Request Response is the most interesting one for the purpose of this tool. The selected operation could have primitive or complex parameters and results. The result of Web Service invocation will be displayed in a table (the tool prompts you to provide labels for the table columns). You can also filter result attributes by specifying an Xpath expression (see `columnOrder` property in the metadata below). This can be useful for complex return types out of which only few attributes are interesting.

The Web Services command line tool supports Web Services with the following binding and encoding styles:

- DOC/literal
- RPC/encoded

#### 5.2.1.1 Security

The tool does not make any invocation of the Web Service and only generates metadata required by Enterprise Manager for target monitoring purposes. Users running this tool are only expected to have read permission on the WSDL file or URL and permission to save generated files to the appropriate directory.

### 5.2.1.2 Syntax

The Web Services command line tool syntax is as follows:

```
wsperfcli [OPTIONS] < WSDL File | URL to WSDL >
```

The `wsperfcli` command accepts the following options:

- **-useSOAP11** Use SOAP version 1.1 instead of default 1.2
- **-useWSIF** Useful for WSDL with SOAP type arrays
- **-usePROXY** Use proxy to connect to Web Service
- **-useSSL** Use SSL to communicate with Web Service

The command line tool optionally accepts a WSDL file name or URL to locate the WSDL for a Web Service. For example, for a card-service Web Service a WSDL URL would be as follows:

```
http://localhost:44861/card-service/card?WSDL
```

The command tool script requires access to the Enterprise Manager home directory (EM\_HOME) to run. The tool defaults to ORACLE\_HOME. The home directory setting can be set using the command line argument `-DEM_HOME`. For example, `-DEM_HOME=/myEMHome`. Other optional arguments are `-DSOAP11` and `-DuseSSL`, which allow metadata to be generated to use an older version of the SOAP protocol and SSL for communication with the Web Service. The option `-useWSIF` is used for WSDL that use the old SOAP style arrays (rpc/encoded Web Services).

The tool parses specified WSDL for all the port types and binding (supported protocols such as HTTP get/post, SOAP) to list all the operations. If there are multiple port types in WSDL, you will first be prompted to choose a port type.

### 5.2.1.3 Usage

The following example shows a sample WSDL file passed to the command line tool to generate the target metadata and collection files.

#### **Example 5-1 Sample WSDL File TargetWithWSMgmtInterface**

```
<?xml version="1.0" encoding="UTF-8" ?>
<definitions
  name="TargetWithWSMgmtInterface"
  targetNamespace="http://tempuri.org"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:tns="http://tempuri.org"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  xmlns:ns1="http://mypackage3/MyProdMgmtInterface.wsdl/types"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  >
  <types>
    <schema xmlns="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://tempuri.org" elementFormDefault="qualified"
      xmlns:tns="http://tempuri.org"
      xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xmlns:soap11-enc="http://schemas.xmlsoap.org/soap/encoding/" />
    <schema xmlns="http://www.w3.org/2001/XMLSchema"
      targetNamespace="http://mypackage3/MyProdMgmtInterface.wsdl/types"
      elementFormDefault="qualified"
```

```

xmlns:tns="http://mypackage3/MyProdMgmtInterface.wsdl/types"
    xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xmlns:soap11-enc="http://schemas.xmlsoap.org/soap/encoding/"
    <complexType name="ArrayOfstring">
        <complexContent>
            <restriction base="soap11-enc:Array">
                <attribute ref="soap11-enc:arrayType"
wsdl:arrayType="string[]" />
            </restriction>
        </complexContent>
    </complexType>
</schema>
</types>
<message name="TargetWithWSMgmtInterfaceSEI1_getNumActiveThreads"/>
<message name="TargetWithWSMgmtInterfaceSEI1_getUserSessionID">
    <part name="user" type="xsd:string"/>
</message>
<message name="TargetWithWSMgmtInterfaceSEI1_getUserSessionIDResponse">
    <part name="result" type="xsd:string"/>
</message>
<message name="TargetWithWSMgmtInterfaceSEI1_getNumActiveThreadsResponse">
    <part name="result" type="xsd:long"/>
</message>
<message name="TargetWithWSMgmtInterfaceSEI1_getActiveUsers"/>
<message name="TargetWithWSMgmtInterfaceSEI1_getActiveUsersResponse">
    <part name="result" type="ns1:ArrayOfstring"/>
</message>
<portType name="ManagementInterfaceWS">
    <operation name="getActiveUsers">
        <input message="tns:TargetWithWSMgmtInterfaceSEI1_getActiveUsers"/>
        <output message="tns:TargetWithWSMgmtInterfaceSEI1_
getActiveUsersResponse"/>
    </operation>
    <operation name="getNumActiveThreads">
        <input message="tns:TargetWithWSMgmtInterfaceSEI1_
getNumActiveThreads"/>
        <output message="tns:TargetWithWSMgmtInterfaceSEI1_
getNumActiveThreadsResponse"/>
    </operation>
    <operation name="getUserSessionID" parameterOrder="user">
        <input message="tns:TargetWithWSMgmtInterfaceSEI1_getUserSessionID"/>
        <output message="tns:TargetWithWSMgmtInterfaceSEI1_
getUserSessionIDResponse"/>
    </operation>
</portType>
<binding name="TargetWithWSMgmtInterfaceSoapHttp"
type="tns:ManagementInterfaceWS">
    <soap:binding style="rpc"
transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="getActiveUsers">
        <soap:operation soapAction="http://tempuri.org:getActiveUsers"/>
        <input>
            <soap:body use="encoded" namespace="http://tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
                parts="" />
        </input>
        <output>
            <soap:body use="encoded" namespace="http://tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"

```

```

        parts="result"/>
    </output>
</operation>
<operation name="getNumActiveThreads">
    <soap:operation soapAction="http://tempuri.org:getNumActiveThreads"/>
    <input>
        <soap:body use="encoded" namespace="http://tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        parts=""/>
    </input>
    <output>
        <soap:body use="encoded" namespace="http://tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        parts="result"/>
    </output>
</operation>
<operation name="getUserSessionID">
    <soap:operation soapAction="http://tempuri.org:getUserSessionID"/>
    <input>
        <soap:body use="encoded" namespace="http://tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        parts="user"/>
    </input>
    <output>
        <soap:body use="encoded" namespace="http://tempuri.org"
encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
        parts="result"/>
    </output>
</operation>
</binding>
<service name="ManagementInterfaceWS">
    <port name="TargetWithWSMgmtInterfacePort"
binding="tns:TargetWithWSMgmtInterfaceSoapHttp">
        <soap:address
location="http://144.25.119.190:8989/WS-MyProdMgmtIntf-context-root/ManagementInte
rfaceWS"/>
    </port>
</service>
</definitions>

```

The following sample Web Services command line tool session uses the WSDL file shown in [Example 5-1](#). First, the tool parses the WSDL for all port types and bindings (supported protocols such as HTTP get/post or SOAP) to list all the operations. If there are multiple port types in the WSDL, the tool will first prompt you to select a port type.

```

All port types for specified WSDL:
0 TargetWithWSMgmtInterfacePort
All operations for specified port type:
0 long getNumActiveThreads()
1 string getUserSessionID(string user)
2 string [] getActiveUsers()
>> Enter the index of operation to select.: 0

```

Next, the tool prompts you to pick attributes from the return type to add to the metric.

```

Return value(s) for operation:
0 /tns:TargetWithWSMgmtInterfaceSEI1_getNumActiveThreadsResponse/result <long>
>> Enter the name for this metric column: ActiveThreads
>> Enter the label for this metric : ActiveThreadsLabel
>> Is this column a key <y/n> [n]: y

```

```

>> Do you want to add another metric <y/n>? [n] :
>> Enter value(s) for operation arguments:
>> Is this metric for periodic collection <y/n>? [n] : y
>> Enter the frequency of collections in seconds:12
>> Do you want to pick another operation to add <y/n>? [n] :n
>> Enter the metadata file name (metadata/TargetWithWSMgmtInterface.xml)..:

```

The command line tool generates the metadata required to monitor the target type TargetWithWSMgmtInterface as shown in [Example 5-2](#).

**Example 5-2 TargetWithWSMgmtInterface Target Metadata File**

```

<?xml version='1.0'?><!DOCTYPE TargetMetadata SYSTEM
"../dtds/TargetMetadata.dtd"><TargetMetadata META_VER="1.0"
TYPE="TargetWithWSMgmtInterface" CATEGORY_PROPERTIES="versionCategory">
<Display>
<Label NLSID="TargetWithWSMgmtInterface">TargetWithWSMgmtInterface</Label>
<ShortName NLSID="TargetWithWSMgmtInterface">TargetWithWSMgmtInterface</ShortName>
<Description
NLSID="TargetWithWSMgmtInterface">TargetWithWSMgmtInterface</Description>
</Display>
<Metric NAME="Threads" TYPE="TABLE">
<Display>
<Label NLSID="Threads">Threads</Label>
</Display>
<TableDescriptor>
<ColumnDescriptor NAME="activeThreads" TYPE="NUMBER">
<Display>
<Label NLSID="ActiveThreads">ActiveThreads</Label>
</Display>
</ColumnDescriptor>
</TableDescriptor>
<QueryDescriptor FETCHLET_ID="OJMX">
<Property NAME="machine" SCOPE="INSTANCE">HTTPMachine</Property>
<Property NAME="port" SCOPE="INSTANCE">HTTPPort</Property>
<Property NAME="metricType" SCOPE="GLOBAL" OPTIONAL="TRUE">GWS</Property>
<Property NAME="metric" SCOPE="GLOBAL">getNumActiveThreads</Property>
<Property NAME="requestBodyElement" SCOPE="GLOBAL"
OPTIONAL="TRUE">tns:getNumActiveThreads</Property>
<Property NAME="authuser" SCOPE="INSTANCE" OPTIONAL="TRUE">authUser</Property>
<Property NAME="authpwd" SCOPE="INSTANCE" OPTIONAL="TRUE">authPasswd</Property>
<Property NAME="documentType" SCOPE="GLOBAL"
OPTIONAL="TRUE">rpc/encoded</Property>
<Property NAME="soapVersion" SCOPE="GLOBAL" OPTIONAL="TRUE">SOAP1.1</Property>
<Property NAME="targetNamespace" SCOPE="GLOBAL"
OPTIONAL="TRUE"><![CDATA[<namespace prefix="tns" uri="http://tempuri.org"
/>]]></Property>
<Property NAME="columnOrder"
SCOPE="GLOBAL">/tns:getNumActiveThreadsResponse/result</Property>
<Property NAME="additionalNamespaces" SCOPE="GLOBAL"
OPTIONAL="TRUE"><![CDATA[<namespaces><namespace prefix="tns"
uri="http://tempuri.org" /><namespace prefix="mime"
uri="http://schemas.xmlsoap.org/wsd/mime/" /><namespace prefix="soap12"
uri="http://schemas.xmlsoap.org/wsd/soap12/" /><namespace prefix="ns1"
uri="http://mypackage3/MyProdMgmtInterface.wsd/types" /><namespace prefix="xsd"
uri="http://www.w3.org/2001/XMLSchema" /><namespace prefix="soap"
uri="http://schemas.xmlsoap.org/wsd/soap/" /></namespaces>]]></Property>
<Property NAME="delimiter" SCOPE="GLOBAL">|</Property>
<Property NAME="URI"
SCOPE="GLOBAL">/WS-MyProdMgmtIntf-context-root/ManagementInterfaceWS</Property>

```

```

<Property NAME="soapAction" SCOPE="GLOBAL"
OPTIONAL="TRUE">http://tempuri.org:getNumActiveThreads</Property>
<Property NAME="name" SCOPE="GLOBAL">getNumActiveThreads</Property>
<Property NAME="returnType" SCOPE="GLOBAL">long</Property>
<Property NAME="arguments"
SCOPE="GLOBAL"><![CDATA[<arguments></arguments>]]></Property>
</QueryDescriptor>
</Metric>
<InstanceProperties>
<InstanceProperty NAME="HTTPMachine"><Display>
<Label NLSID="dms_HTTPMachine_iprop">Machine name</Label>
<ShortName NLSID="dms_HTTPMachine_iprop">Machine name</ShortName>
</Display>
</InstanceProperty>
<InstanceProperty NAME="HTTPPort"><Display>
<Label NLSID="dms_HTTPPort_iprop">Port</Label>
<ShortName NLSID="dms_HTTPPort_iprop">Port</ShortName>
</Display>
</InstanceProperty>
<InstanceProperty NAME="version" OPTIONAL="TRUE"><Display>
<Label NLSID="oc4j_version_iprop">Version of TargetWithWSMgmtInterface</Label>
<ShortName NLSID="oc4j_version_iprop">Version of
TargetWithWSMgmtInterface</ShortName>
</Display>
</InstanceProperty>
<InstanceProperty NAME="authUser" OPTIONAL="TRUE"><Display>
<Label NLSID="dms_authUser_iprop">Username for basic authorization</Label>
<ShortName NLSID="dms_authUser_iprop">Username for basic authorization</ShortName>
</Display>
</InstanceProperty>
<InstanceProperty NAME="authPasswd" OPTIONAL="TRUE" CREDENTIAL="TRUE"><Display>
<Label NLSID="dms_authPasswd_iprop">Password for basic authorization</Label>
<ShortName NLSID="dms_authPasswd_iprop">Password for basic
authorization</ShortName>
</Display>
</InstanceProperty>
</InstanceProperties>
</TargetMetadata>

```

The Command Line Tool also generates the requisite collection file as shown in

### **Example 5-3 TargetWithWSMgmtInterface Default Collection File**

```

<!DOCTYPE TargetMetadata SYSTEM "../dtds/TargetCollection.dtd"><!-- This file is
generated by Collector at 2006-01-24 14:00:10 -->

<TargetCollection TYPE="TargetWithWSMgmtInterface"
NAME="TargetWithWSMgmtInterface" INCLUDE_DEFAULT="TRUE">
  <CollectionItem NAME="Threads" UPLOAD="5">
    <Schedule>
      <IntervalSchedule INTERVAL="12" TIME_UNIT="Sec"/>
    </Schedule>
  </CollectionItem>

</TargetCollection>

```

## 5.3 Creating a Management Plug-in Archive

Using the Enterprise Manager Command Line Interface (EM CLI), you create a Management Plug-in Archive, as shown in the following example. META\_FILE and

DC\_FILE represent the target metadata and default collection files generated by the Web Services Command Line tool.

```
> ./emcli add_mp_to_mpa -mpa=TargetWithWSMgmtInterface.jar -mp_version="1.0"
-ttd="{META_FILE}" -dc="{DC_FILE}"
```

For more information about the EM CLI, see *Oracle Enterprise Manager Command Line Interface*.

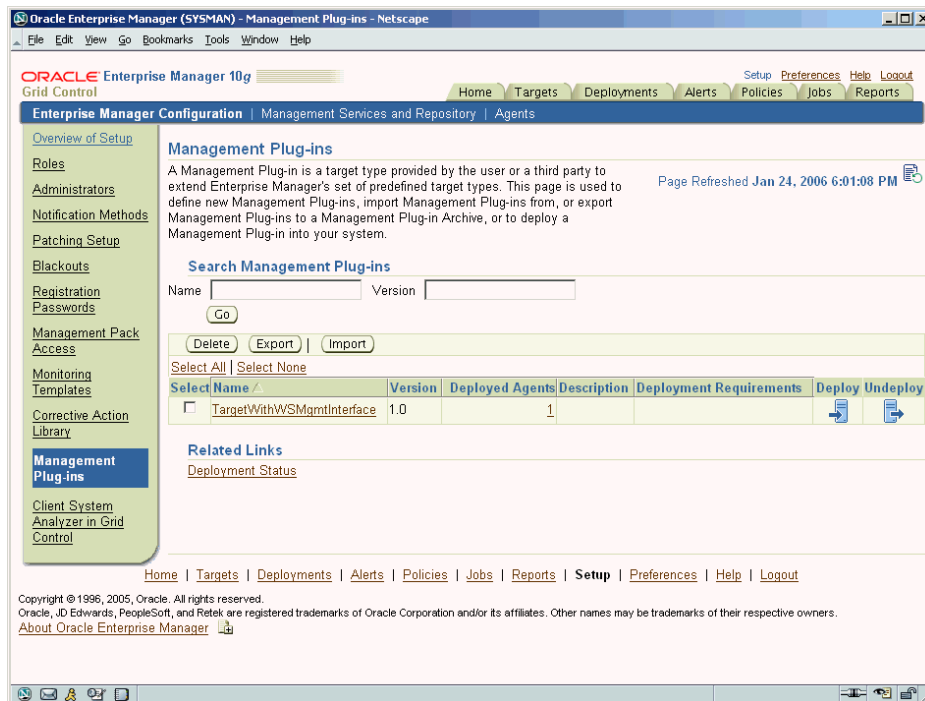
Once the archive jar file is created, you are ready to upload the archive to Enterprise Manager and import the Management Plug-in. For more information about Management Plug-in Archives, see [Chapter 2, "Developing a Management Plug-in"](#).

## 5.4 Importing the Management Plug-in

The next step is to import the Management Plug-in from the archive file. This operation is performed from the Enterprise Manager console.

1. From the Enterprise Manager console, click Setup.
2. Click Management Plug-ins from the left navigation bar.
3. Click Import.
4. From the Import page, specify the Management Plug-in Archive file and click List Archive. All Management Plug-ins contained within the archive are displayed.
5. Select the Web service Management Plug-in from the list and click OK. Enterprise Manager returns you to the Management Plug-ins page with your Web service-specific plug-in added to the list.

**Figure 5–1 Importing the TargetWithWSMgmtInterface Management Plug-in**



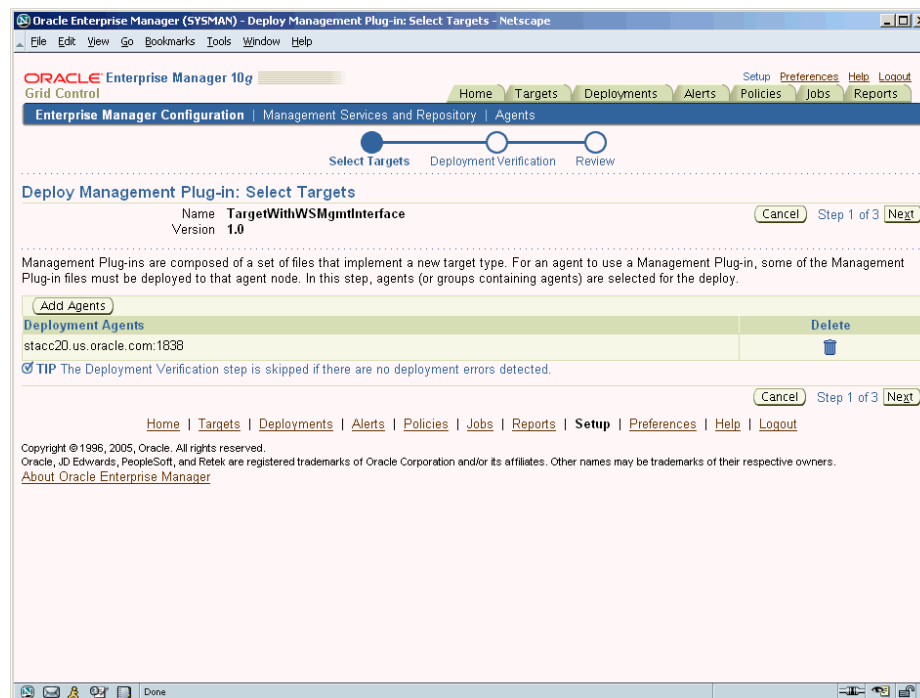


## 5.5 Deploying the Management Plug-in to the Management Agent

Once the Management Plug-in has been imported into Enterprise Manager, you are now ready to deploy the plug-in to any number of Management Agents. Deploying the Management Plug-in to an Agent allows that Agent to monitor targets exposing a Web Service management interface.

1. From the Management Plug-in main page, click the Deploy icon for the Management Plug-in you wish to deploy. The Deploy Management Plug-in: Select Targets page appears, as shown in the following figure.

**Figure 5–2 Deploying the Management Plug-in to Management Agents**



2. Click Add Agents. The Search and Select: Agents page appears.
3. Choose the desired Management Agent. Use the appropriate search parameters to find the desired Agent.
4. Click Select. Enterprise Manager returns you to the Deploy Management Plug-in: Select Targets page. The selected Agent(s) appears in the deployment list.

If deployment issues are detected for one or more the selected Agents, the Deployment Verification page is displayed. Only Agents for which an issue is detected appear in the table on this page. See online help for more information.

5. Click Next to go to the Review page. The Management Plug-in name and version are shown. This page contains important information pertaining to the deployment process. Ensure that you understand the information before proceeding.
6. Click Finish.

## 5.6 Adding a Target Instance

Now that the Management Plug-in has been deployed to the appropriate Agent(s), you are ready to begin monitoring targets that expose a Web Service management interface, as defined by your Management Plug-in.

1. From the Monitored Targets section of the Management Agent home page, choose the desired target type from the **Add** drop-down menu and click **Go**. The Add target page appears.

Note that **Username/Password for basic authorization** are the login credentials used to access the Web Service.

2. Enter the requisite target properties and click OK (shown in [Figure 5-3](#)). The newly added target appears in the Agent's Monitored Targets list.

**Figure 5-3 Adding a Target Instance**

Oracle Enterprise Manager (SYSMAN) - Add TargetWithWSMgmtInterface - Netscape

ORACLE Enterprise Manager 10g

Grid Control

Home Targets Deployments Alerts Policies Jobs Reports

Enterprise Manager Configuration | Management Services and Repository | Agents

Add TargetWithWSMgmtInterface

Cancel OK

**Properties**

\* Name

Type TargetWithWSMgmtInterface

Name	Value
Machine name	localhost
Port	8888
Version of TargetWithWSMgmtInterface	1
Username for basic authorization	mywslogon
Password for basic authorization	*****

**Monitoring**

Oracle has automatically enabled monitoring for this target's availability and performance, so no further monitoring configuration is necessary. You can edit the metric thresholds from the target's homepage.

Cancel OK

Home | Targets | Deployments | Alerts | Policies | Jobs | Reports | Setup | Preferences | Help | Logout

Copyright © 1996, 2005, Oracle. All rights reserved.  
Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
[About Oracle Enterprise Manager](#)

## 5.7 Viewing Monitored Metrics

With a target instance added to the Agent for monitoring, you are now ready to view metrics defined for your target type.

1. From the Agent home page, click on the target instance you added in the previous step in the **Monitored Targets** list. Enterprise Manager takes you to that target's home page.
2. In the **Related Links** section, click **All Metrics**. The All Metrics page appears for the monitored target. An expandable tree list for each metric allows you to drill down to view specific metric parameters, as show in [Figure 5-4](#).

Figure 5-4 Viewing Metrics

The screenshot shows the Oracle Enterprise Manager 10g interface in a Netscape browser window. The page title is "Oracle Enterprise Manager (SYSMAN) - All Metrics - Netscape". The browser's address bar shows the URL "TargetWithWSMgmtInterface: MyWSMgmtInterface >". The page content includes a navigation menu with "Home", "Targets", "Deployments", "Alerts", "Policies", "Jobs", and "Reports". Below the navigation menu, the "All Metrics" section is displayed, showing a table of metrics for the target "MyWSMgmtInterface". The table has columns for "Metrics", "Thresholds", "Collection Schedule", "Upload Interval", and "Last Upload". The metrics listed are "Threads" (with a sub-entry "ActiveThreads") and "ActiveThreads". The "Threads" metric has a threshold of "None", a collection schedule of "Every 12 Seconds", and an upload interval of "Every 5 Collections". The "ActiveThreads" metric has a threshold of "Not Set". Below the table, there are "Related Links" and a footer with copyright information.

Expand All | Collapse All

Collected From Target

Metrics	Thresholds	Collection Schedule	Upload Interval	Last Upload
MyWSMgmtInterface				
Threads	None	Every 12 Seconds	Every 5 Collections	-
ActiveThreads	Not Set			

Related Links

[Metric and Policy Settings](#)

[Home](#) | [Targets](#) | [Deployments](#) | [Alerts](#) | [Policies](#) | [Jobs](#) | [Reports](#) | [Setup](#) | [Preferences](#) | [Help](#) | [Logout](#)

Copyright © 1996, 2005, Oracle. All rights reserved.  
 Oracle, JD Edwards, PeopleSoft, and Retek are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.  
[About Oracle Enterprise Manager](#)



# Part II

---

## Reference

This section of the guide provides reference information to be used while developing Management Plug-ins.

Part II contains the following chapters:

- [Chapter 6, "Management Repository Views"](#)
- [Chapter 7, "Fetchlets"](#)



---

---

# Management Repository Views

Enterprise Manager repository views are used to access information in the Management Repository for further processing and/or presentation.

---

---

**Important:** Views supplied with Enterprise Manager 10g release 10.2 are provisional. Provisional views, although fully supported for the current release of Enterprise Manager 10g, may change in subsequent releases. Backward compatibility is not guaranteed for provisional views.

---

---

This chapter covers the following:

- [Overview](#)
- [Monitoring Views](#)
- [Inventory Views](#)
- [Policy Definition Views](#)
- [Policy Association Views](#)
- [Policy Violation Views](#)
- [Management Template Views](#)
- [Job Views](#)
- [Application Service Level Management Views](#)
- [Configuration Views](#)
- [Oracle Home Patching Views](#)
- [Linux Patching Views](#)
- [Security Views](#)
- [Configuration Management Views](#)
- [Database Cluster Views](#)
- [Storage Reporting Views](#)

## 6.1 Overview

The Enterprise Manager Management Repository views provide access to target, metric, and monitoring information stored in the Management Repository. Accessing the repository will allow you to perform the following:

- Obtain pertinent application-specific information at the right level of granularity and density for a wider variety of users: IT staff, executives, developers.
- Send alerts for metric threshold violations.
- Perform historical analysis or additional computation on stored data.
- Seamless integration of Enterprise Manager alerts with user ticketing systems, such as iSupport and Remedy.

The Management Repository is the comprehensive source for all the management information for Enterprise Manager, with the key to extensibility being the repository's open schema. This open architecture allows users to customize how the information in the repository is used if Enterprise Manager's standard configuration does not meet their requirements. To facilitate easy access to information stored in the repository, Enterprise Manager supplies a comprehensive set of views rather than forcing the user to access repository base tables directly. Views buffer custom applications from any underlying changes to the repository schema and ensures up-stream applications will not break when the repository schema changes via patching or new releases.

### Using Repository Views

Because the views are simple queries to a database, users can imbed these queries within any application code used to return information for further processing and/or display in the Enterprise Manager Grid Control console.

As shown in [Example 6-1, "View Usage"](#), the Java code uses Enterprise Manager views to query the Management Repository rather than accessing the repository tables directly. For each of four time windows, there are four SQL statements with question marks (?) as placeholders for the parameters.

#### Example 6-1 View Usage

```
public static final String hour_stmt =
"SELECT collection_timestamp, value "+
"FROM mgmt$metric_details " +
"WHERE target_type = ? and target_name = ? and metric_name = ? and metric_column=
? " +
"and collection_timestamp > sysdate - 1/24 " +
"ORDER BY collection_timestamp ";

public static final String day_stmt =
"SELECT rollup_timestamp, average "+
"FROM mgmt$metric_hourly " +
"WHERE target_type = ? and target_name = ? and metric_name = ? and metric_column=
? " +
"and rollup_timestamp > sysdate - 1 " +
"ORDER BY rollup_timestamp";

public static final String week_stmt =
"SELECT rollup_timestamp, average "+
"FROM mgmt$metric_daily " +
"WHERE target_type = ? and target_name = ? and metric_name = ? and metric_column=
? " +
"and rollup_timestamp > sysdate - 7 " +
"ORDER BY rollup_timestamp";

public static final String month_stmt =
"SELECT rollup_timestamp, average "+
"FROM mgmt$metric_daily " +
```



```
"WHERE target_type = ? and target_name = ? and metric_name = ? and metric_column=
? " +
"and rollup_timestamp > sysdate - 31 " +
"ORDER BY rollup_timestamp";
```

## 6.2 Monitoring Views

### 6.2.1 MGMT\$METRIC\_ERROR\_HISTORY

MGMT\$METRIC\_ERROR\_HISTORY displays the history of metric collection errors that have occurred in last 3 months.

**Table 6–1 MGMT\$METRIC\_ERROR\_HISTORY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	The name of the target for which the metric collection error occurred.
TARGET_TYPE	VARCHAR2(64)	Target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	Target GUID of the target for which the metric collection error occurred.
METRIC_NAME	VARCHAR2(64)	Name of the metric for which the error occurred.
METRIC_LABEL	VARCHAR2(64)	User display name of the metric for which the error occurred.
COLL_NAME	VARCHAR2(256)	Name of the collection collecting the metric for which the error occurred.
COLLECTION_TIMESTAMP	DATE	Time when the collection error occurred.
ERROR_TYPE	NUMBER	Type of metric error. 0=Error 1=Warning
ERROR_MESSAGE	VARCHAR2(4000)	Text of the error message.

#### Usage Notes

List the history of metric errors for a given target.

List the history of metric errors for a given target and metric.

Queries using this view would use index if target\_name and target\_type were specified.

### 6.2.2 MGMT\$METRIC\_ERROR\_CURRENT

MGMT\$METRIC\_ERROR\_CURRENT displays the metric collections that are currently generating errors.

**Table 6–2 MGMT\$METRIC\_ERROR\_CURRENT**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target for which the metric collection error occurred.

**Table 6–2 (Cont.) MGMT\$METRIC\_ERROR\_CURRENT**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	Target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	GUID of the target for which the metric collection error occurred.
METRIC_NAME	VARCHAR2(64)	Name of the metric for which the error occurred
METRIC_LABEL	VARCHAR2(64)	User display name of the metric for which the error occurred.
COLL_NAME	VARCHAR2(256)	Name of the collection collecting the metric for which the error occurred
COLLECTION_TIMESTAMP	DATE	Time when the collection error occurred.
ERROR_TYPE	NUMBER	Type of metric error. 0=Error 1=Warning
ERROR_MESSAGE	VARCHAR2(4000)	Text of the error message.

**Usage Notes**

List the current metric errors for a given target.

List the current metric errors for a given target and metric.

Queries using this view would use index if target\_name and target\_type were specified.

**6.2.3 MGMT\$TARGET\_COMPONENTS**

MGMT\$TARGET\_COMPONENTS displays information about the software components that are associated with a managed target as well as details about where the software components have been installed.

**Table 6–3 MGMT\$TARGET\_COMPONENTS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	Target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier (GUID) for the target.
HOST_NAME	VARCHAR2(64)	The name of the host where the software homes are located.
HOME_NAME	VARCHAR2(64)	The name for the location where the software is installed. This might be a user-supplied name or a name supplied by the Oracle Universal Installer.

**Table 6–3 (Cont.) MGMT\$TARGET\_COMPONENTS**

Column	Datatype	Description
HOME_TYPE	VARCHAR2(64)	Three categories of homes are available: · ORACLE_HOME · APPL_TOP · INDEPENDENT · UNKNOWN
HOME_LOCATION	VARCHAR2(128)	The file system path to the root of the home.
COMPONENT_NAME	VARCHAR2(128)	The name of the software component.
COMPONENT_EXTERNAL_NAME	VARCHAR2(128)	The external name of the target, as specified during installation.
COMPONENT_VERSION	VARCHAR2(64)	The current version of the software component. If the software component has been patched, this version number may be different than the base version number.
COMPONENT_BASE_VERSION	VARCHAR2(64)	Base version of the software component.
SNAPSHOT_GUID	RAW(16)	GUID of the snapshot.

**Usage Notes**

This view can be used to identify the software components that are part of a specific managed target.

Access to this view will use an index if the query references the target name and target type, the home location, or the host name.

**6.2.4 MGMT\$BLACKOUT\_HISTORY**

MGMT\$BLACKOUT\_HISTORY displays a historical log of changes in the blackout state for a managed target. In addition, the view can be used to generate a list of targets that were in a blackout period for a specific period of time.

**Table 6–4 MGMT\$BLACKOUT\_HISTORY**

Column	Datatype	Description
BLACKOUT_NAME	VARCHAR(64)	The name of the blackout.
CREATED_BY	VARCHAR(256)	The Enterprise Manager administrator who created the blackout.
BLACKOUT_GUID	RAW(16)	The unique global identifier (GUID) for the blackout.
START_TIME	DATE	Start of the blackout period for the managed target.
END_TIME	DATE	End of the blackout period for the managed target. If the target is currently in a blackout period, the END_TIMESTAMP date will be NULL.
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.

**Table 6–4 (Cont.) MGMT\$BLACKOUT\_HISTORY**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	Types of targets may include databases, hosts, web servers, applications, or Application Servers. The definer of the collection definition at the Management Agent defines the target type. The target type defines the set of metrics that are collected for a managed target within the Management Repository.
STATUS	NUMBER(2)	Current status of the blackout. Status Values: 0: Scheduled 1: Start Processing 2: Start Partial 4: Started 5: Stop Pending 6: Stop Failed 7: Stop Partial 8: Edit Failed 9: Edit Partial 10: Stopped 11: Ended 12: Partial Blackout 13: Modify Pending

**Usage Notes**

Queries using this view will use an index if they reference the target\_name, target\_type, start\_timestamp, or end\_timestamp.

Typically, blackout history information retrieved using this view will be ordered by target\_name, target\_type, and start\_timestamp.

**6.2.5 MGMT\$BLACKOUTS**

MGMT\$BLACKOUTS displays all blackout definition information along with current schedules.

**Table 6–5 MGMT\$BLACKOUTS**

Column	Datatype	Description
BLACKOUT_NAME	VARCHAR(64)	The name of the blackout.
BLACKOUT_GUID	RAW(16)	The unique global identifier (GUID) of the blackout.
REASON	VARCHAR(64)	Purpose of the blackout. Reasons are chosen from a predefined list by the report owner.
DESCRIPTION	VARCHAR2(2000)	Detailed information about the blackout.

**Table 6–5 (Cont.) MGMT\$BLACKOUTS**

Column	Datatype	Description
STATUS	NUMBER(2)	Current status of the blackout. Status Values: 0: Scheduled 1: Start Processing 2: Start Partial 4: Started 5: Stop Pending 6: Stop Failed 7: Stop Partial 8: Edit Failed 9: Edit Partial 10: Stopped 11: Ended 12: Partial Blackout 13: Modify Pending
CREATED_BY	VARCHAR(256)	Administrator who created the blackout. CREATED_BY returns SYSTEM as the blackout owner if the blackout was created using the Enterprise Manager Command Line Interface.
LAST_START_TIME	DATE	Last time the blackout successfully started.
LAST_END_TIME	DATE	Last time the blackout successfully ended.
SCHEDULED_TIME	NUMBER	Possible values are: 0 - Immediate schedule 1 - Run once at specified time 2 - Run on interval 3 - Run daily 4 - Run on specified days of the week 5 - Run on specified days of the month 6 - Run on specified days of the year
SCHEDULE_START_TIME	DATE	Time the blackout is scheduled to start.
SCHEDULE_END_TIME	DATE	Time the blackout is scheduled to end.
DURATION	NUMBER	Duration of the blackout in minutes.

## 6.2.6 MGMT\$ALERT\_ANNOTATIONS

MGMT\$ALERT\_ANNOTATIONS displays a summary of alert annotations.

**Table 6–6 MGMT\$**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator
TARGET_TYPE	VARCHAR2(64)	Target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier (GUID) for the target.

**Table 6–6 (Cont.) MGMT\$**

Column	Datatype	Description
METRIC_NAME	VARCHAR2(64)	Name of the metric.
METRIC_COLUMN	VARCHAR2(64)	If the metric defined at collection time is a TABLE type, the name of the column in the table that is represented by this metric is written to this field. The metric column is part of the primary key that uniquely identifies the value of this table that this metric represents.
METRIC_LABEL	VARCHAR2(64)	Display name for the metric.
COLUMN_LABEL	VARCHAR2(64)	Display name for the column of a TABLE metric.
KEY_VALUE	VARCHAR2(256)	Key value of the alert violation. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key.
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key.
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key.
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fourth part of the key.
MESSAGE	VARCHAR2(4000)	The metric severity message.
ALERT_STATE	VARCHAR2(20)	A user readable description of the internal alert code that is sent from the Management Agent to identify the state of the alert condition. An alert record is transferred to the repository from the Management Agent each time the metric threshold is crossed in either direction, or if the Management Agent is restarted. The value of this column will contain one of the following strings: <ul style="list-style-type: none"> <li>· Clear</li> <li>· Warning</li> <li>· Critical</li> </ul>
COLLECTION_TIMESTAMP	DATE	Date and time when the alert condition was detected by the Management Agent.
ANNOTATION_MESSAGE	VARCHAR2(4000)	Annotation text.
ANNOTATION_TIMESTAMP	DATE	Time the annotation was created.
ANNOTATED_BY	VARCHAR2(64)	Enterprise Manager administrator who authored the the annotation.
SOURCE_OBJ_GUID	RAW(16)	Unique global identifier (GUID) of the metric severity associated with this annotation.

## 6.2.7 MGMT\$ALERT\_NOTIF\_LOG

MGMT\$ALERT\_NOTIF\_LOG displays a summary of alert notifications.

**Table 6–7 MGMT\$MGMT\$ALERT\_NOTIF\_LOG**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier (GUID) for the target.
METRIC_NAME	VARCHAR2(64)	The name of the metric being defined.
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.  For example, if the table describing the MGMT\$TARGET_TYPE view was being defined as a table metric, Column Name, Data Type, and Description would be metric columns.
METRIC_LABEL	VARCHAR2(64)	A intuitive display name for the metric that is being defined.
COLUMN_LABEL	VARCHAR2(64)	For table metrics, the column label contains a user understandable display name for the metric column.
KEY_VALUE	VARCHAR2(256)	For table metrics, this column contains the value of the key column for the row in the table whose thresholds are being defined. If the thresholds are not for a table metric, or the thresholds apply for all rows in the metric column, then the value in this column will contain a single space.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key.
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key.
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key.
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key.
MESSAGE	VARCHAR2(4000)	An optional message that is generated when the alert is created that provides additional information about the alert condition.
ALERT_STATE	VARCHAR2(20)	A user readable description of the internal alert code that is sent from the Management Agent to identify the state of the alert condition. A alert record is transferred to the repository from the Management Agent each time the metric threshold is crossed in either direction, or if the Management Agent is restarted. The value of this column will contain one of the following strings:  · Clear · Warning · Critical

**Table 6–7 (Cont.) MGMT\$MGMT\$ALERT\_NOTIF\_LOG**

Column	Datatype	Description
COLLECTION_TIMESTAMP	DATE	Date and time when the alert condition was detected by the Management Agent.
DELIVERY_MESSAGE	VARCHAR2(1024)	Message indicating the success or failure of the notification delivery.
DELIVERY_TIMESTAMP	DATE	Time at which the log message was created.
SOURCE_OBJ_GUID	RAW(16)	Unique global identifier (GUID) of the metric severity associated with this delivery message.

## 6.2.8 MGMT\$TARGET\_METRIC\_COLLECTIONS

MGMT\$TARGET\_METRIC\_COLLECTIONS displays information about the metric collections.

**Table 6–8 MGMT\$TARGET\_METRIC\_COLLECTIONS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	Unique global identifier (GUID) for the target.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.  For example, if the table describing the MGMT\$TARGET_TYPE view was being defined as a table metric, Column Name, Data Type, and Description would be metric columns.
METRIC_GUID	RAW(16)	Unique global identifier (GUID) for the metric. This ID can be used to associate metric information with metric data information during reporting.
COLLECTION_NAME	VARCHAR2(64)	The name of the collection.
IS_ENABLED	NUMBER(1)	Indicates whether the collection is currently enabled.  0=not enabled 1=enabled.
IS_REPOSITORY	NUMBER(1)	Indicates whether this is a repository-side collection. A repository-side collection has a PL/SQL evaluation procedure that is responsible for calculating the metric values.



**Table 6–8 (Cont.) MGMT\$TARGET\_METRIC\_COLLECTIONS**

Column	Datatype	Description
FREQUENCY_CODE	NUMBER	The metric collection frequency type. Possible values are: 1: One Time 2: Interval 3: Daily 4: Weekly 5: Monthly 6: Yearly 7: On Demand
COLLECTION_FREQUENCY	VARCHAR2	Frequency of the metric collection. Value displayed is dependent on the frequency code: <sup>2</sup> For One Time, the start date-time is stored in DD-MON-YY HH24:MI format. <sup>2</sup> For Interval type, the frequency in minutes is stored. <sup>2</sup> For Daily/Weekly/Monthly/Yearly types, the hour and minute of collection is stored in HH24:MI format. <sup>2</sup> For On-Demand type, On-Demand is stored.
UPLOAD_POLICY	NUMBER	The frequency with which the metric data is uploaded/stored.

**Usage Notes**

List the metric collections for a given target.

**6.2.9 MGMT\$TARGET\_METRIC\_SETTINGS**

MGMT\$TARGET\_METRIC\_SETTINGS displays information about the current metric setting stored for all targets in the Management Repository. This view provides information for both agent-side and repository-side metrics.

**Table 6–9 MGMT\$TARGET\_METRIC\_SETTINGS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier (GUID) for the target.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.

**Table 6–9 (Cont.) MGMT\$TARGET\_METRIC\_SETTINGS**

Column	Datatype	Description
METRIC_COLUMN	VARCHAR2(64)	<p>For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.</p> <p>For example, if the table describing the MGMT\$TARGET_TYPE view was being defined as a table metric, Column Name, Data Type, and Description would be metric columns.</p>
METRIC_GUID	RAW(16)	The unique global identifier for the metric. This ID can be used to associate metric information with metric data information during reporting.
COLLECTION_NAME	VARCHAR2(64)	The name of the collection.
CATEGORY	VARCHAR2(64)	<p>The name of the category the metric.</p> <p>Refer to MGMT\$METRIC_CATEGORIES for the list of all metric categories.</p>
KEY_VALUE	VARCHAR2(256)	<p>The key value of the metric setting. For composite keys, this is the first part of the key.</p> <p>If the thresholds are not for a table metric, or the thresholds apply for all rows in the metric column, then the value in this column will contain a single space.</p>
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key.
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key.
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key.
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key.
KEY_OPERATOR	NUMBER	<p>Specifies whether the key_value columns have any SQL wildcards.</p> <p>For single key column metrics, the value is 1 if the key_value has wildcard characters, 0 otherwise.</p> <p>For metrics with multiple keys, a list of operators for all key columns will be stored here. For example, a metric with 3 keys (k1, k2, k3) where K1 and K2 use wildcards and K3 uses exact match, then 011 is stored in this column.</p>
HAS_ACTIVE_BASELINE	NUMBER(1)	The is a flag that specifies that the metric rows with this key_value has an active baseline and any user updates to thresholds/parameter values should be ignored.
PREVENT_OVERRIDE	NUMBER(1)	The is a flag that specifies that the metric rows with this key_value has a template override flag. Once the template override flag is ON, any template application will not update the threshold/parameter values.

**Table 6–9 (Cont.) MGMT\$TARGET\_METRIC\_SETTINGS**

Column	Datatype	Description
WARNING_OPERATOR	NUMBER	Defines the warning threshold condition to be applied 0 - GT 1 - EQ 2 - LT 3 - LE 4 - GE 5 - CONTAINS 6 - NE 7 - MATCH : regular expression
WARNING_THRESHOLD	VARCHAR2(256)	The warning threshold value.
CRITICAL_OPERATOR	NUMBER	Defines the critical threshold condition to be applied. 0 - GT 1 - EQ 2 - LT 3 - LE 4 - GE 5 - CONTAINS 6 - NE 7 - MATCH : regular expression
CRITICAL_THRESHOLD	VARCHAR2(256)	The critical threshold value.
OCCURRENCE_COUNT	NUMBER	The number of times the test has to trigger to raise a violation.
WARNING_ACTION_TYPE	VARCHAR2(32)	Specifies the job type of the warning corrective action when WARNING_ACTION_TYPE is "Corrective-Action".
WARNING_ACTION_JOB_OWNER	VARCHAR2(256)	Specifies the job owner of the warning corrective action when WARNING_ACTION_TYPE is "Corrective-Action"
WARNING_ACTION_JOB_NAME	VARCHAR2(64)	Specifies the job name of the warning corrective action when WARNING_ACTION_TYPE is "Corrective-Action"
CRITICAL_ACTION_TYPE	VARCHAR2(17)	The critical corrective action type configured. Possible values are : No-Action : when no action is configured Corrective-Action : when a repository side corrective action is configured Agent-Fixit Job : when an agent side fix-it job is configured.
CRITICAL_ACTION_JOB_TYPE	VARCHAR2(32)	Specifies the job type of the critical corrective action when WARNING_ACTION_TYPE is "Corrective-Action".
CRITICAL_ACTION_JOB_OWNER	VARCHAR2(256)	Specifies the job owner of the critical corrective action when WARNING_ACTION_TYPE is "Corrective-Action".
CRITICAL_ACTION_JOB_NAME	VARCHAR2(64)	Specifies the job name of the critical corrective action when WARNING_ACTION_TYPE is "Corrective-Action".

**Usage Notes**

List all the metric setting for a given target.

List the metric settings for a given target and metric.

List the corrective actions assigned for a given target-metric.

**6.2.10 MGMT\$AVAILABILITY\_CURRENT**

MGMT\$AVAILABILITY\_CURRENT displays information about the most recent target availability information stored in the Management Repository.

**Table 6–10 MGMT\$AVAILABILITY\_CURRENT**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
START_TIMESTAMP	DATE	The time when the target availability status change was first detected.
AVAILABILITY_STATUS	VARCHAR(15)	Current target availability status. This column contains one of the following values: <sup>2</sup> Target Down <sup>2</sup> Target Up <sup>2</sup> Metric Error <sup>2</sup> Agent Down <sup>2</sup> Unreachable <sup>2</sup> Blackout <sup>2</sup> Pending/Unknown

**Usage Notes**

Get the current availability status of a given target.

**6.2.11 MGMT\$AVAILABILITY\_HISTORY**

MGMT\$AVAILABILITY\_HISTORY displays detailed historical information about changes in the availability status for a target over time.

**Table 6–11 MGMT\$AVAILABILITY\_HISTORY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.

**Table 6–11 (Cont.) MGMT\$AVAILABILITY\_HISTORY**

Column	Datatype	Description
START_TIMESTAMP	DATE	The time when the target availability status change was first detected.
END_TIMESTAMP	DATE	The time when the target availability status change was last detected.
AVAILABILITY_STATUS	VARCHAR2(10)	Target availability status. This column will contain one of the following values: <sup>2</sup> Target Down <sup>2</sup> Target Up <sup>2</sup> Metric Error <sup>2</sup> Agent Down <sup>2</sup> Unreachable <sup>2</sup> Blackout <sup>2</sup> Pending/Unknown

**Usage Notes**

Access to this view will use an index if the query references the member TARGET\_NAME, TARGET\_TYPE and the START\_TIMESTAMP.

**6.2.12 MGMT\$ALERT\_CURRENT**

MGMT\$ALERT\_CURRENT displays current information for any alerts that are logged in the Management Repository that are in a non-clear state. Only the most recent open alert in a non-clear status for a given metric will be displayed through this view.

**Table 6–12 MGMT\$**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
VIOLATION_GUID	RAW(16)	Unqie identifier for the alert.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.  For example, if this table describing the MGMT\$TARGET_TYPE view was being defined as a table metric, Column Name, Data Type, and Description would be metric columns.
METRIC_LABEL	VARCHAR2(64)	An intuitive display name for the metric that is being defined.
COLUMN_LABEL	VARCHAR2(64)	For table metrics, the column label contains an intuitive display name for the metric column.

**Table 6–12 (Cont.) MGMT\$**

Column	Datatype	Description
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
COLLECTION_TIMESTAMP	DATE	The date-time when the alert condition was detected by the Management Agent.
ALERT_STATE	VARCHAR2(20)	A user readable description of the internal alert code that is sent from the Management Agent to identify the state of the alert condition. A alert record is transferred to the repository from the Management Agent each time the metric threshold is crossed in either direction, or if the Management Agent is restarted. The value of this column will contain one of the following strings: <ul style="list-style-type: none"> <li>· Warning</li> <li>· Critical</li> </ul> If the metric's alert condition goes into a clear state, it will no longer be visible from this view.
VIOLATION_TYPE	VARCHAR2(19)	A user readable description of the type of violation. Possible values are : Threshold Violation, when the alert is triggered based on a metric threshold Availability, when the alert is triggered for an availability metric Policy Violation, when the alert is triggered based on a policy violation.
MESSAGE	VARCHAR2(1024)	An optional message that is generated when the alert is created that provides additional information about the alert condition.
MESSAGE_NLSID	VARCHAR2(64)	The nlsid of the alert message
MESSAGE_PARAMS	VARCHAR2(4000)	Contains the URL encoded parameters separated by "&" to be used to format the alert message.
ACTION_MESSAGE	VARCHAR2(4000)	Suggested action message in english for this alert
ACTION_MESSAGE_NLSID	VARCHAR2(64)	The NLS ID of the action message.
ACTION_MESSAGE_PARAMS	VARCHAR2(4000)	Contains the URL encoded parameters for translating action message
TYPE_DISPLAY_NAME	VARCHAR2(128)	The display name of the target type

**Usage Notes**

List the current alerts that are in a non-clear state for a metric, set of metrics, or for a managed target. If the user is only interested in non-clear alerts, counts or selects, using this view provide better performance than using the MGMT\$ALERT\_DETAILS view.

Access to this view will use an index if the query references the member target name, target type, metric name, and metric column or a subset of these columns if they are included as listed above from left to right.

### 6.2.13 MGMT\$ALERT\_HISTORY

MGMT\$ALERT\_HISTORY displays historical information for any alerts that are logged in the Management Repository.

**Table 6-13 MGMT\$ALERT\_HISTORY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
VIOLATION_GUID	RAW(16)	Unqiue identifier for the alert.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.  For example, if this table describing the MGMT\$TARGET_TYPE view was being defined as a table metric, Column Name, Data Type, and Description would be metric columns.
METRIC_LABEL	VARCHAR2(64)	An intuitive display name for the metric that is being defined.
COLUMN_LABEL	VARCHAR2(64)	For table metrics, the column label contains an intuitive display name for the metric column.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
COLLECTION_TIMESTAMP	DATE	The date-time when the alert condition was detected by the Management Agent.

**Table 6–13 (Cont.) MGMT\$ALERT\_HISTORY**

Column	Datatype	Description
ALERT_STATE	VARCHAR2(20)	A user readable description of the internal alert code that is sent from the Management Agent to identify the state of the alert condition. A alert record is transferred to the repository from the Management Agent each time the metric threshold is crossed in either direction, or if the Management Agent is restarted. The value of this column will contain one of the following strings: · Warning · Critical If the metric's alert condition goes into a clear state, it will no longer be visible from this view.
ALERT_DURATION	NUMBER	The time, in hours, from when the alert condition was first detected until it was cleared.
MESSAGE	VARCHAR2(1024)	An optional message that is generated when the alert is created that provides additional information about the alert condition.
MESSAGE_NLSID	VARCHAR2(64)	The nlsid of the alert message
MESSAGE_PARAMS	VARCHAR2(4000)	Contains the URL encoded parameters separated by "&" to be used to format the alert message.
ACTION_MESSAGE	VARCHAR2(4000)	Suggested action message in english for this alert
ACTION_MESSAGE_NLSID	VARCHAR2(64)	The NLS ID of the action message.
ACTION_MESSAGE_PARAMS	VARCHAR2(4000)	Contains the URL encoded parameters for translating action message
VIOLATION_TYPE	VARCHAR2(19)	An intuitive description of the type of violation. Possible values are : Threshold Violation: When the alert is triggered based on a metric threshold Availability: When the alert is triggered for an availability metric Policy Violation: When the alert is triggered based on a policy violation.
TYPE_DISPLAY_NAME	VARCHAR2(128)	The display name of the target type

**Usage Notes**

List the historical details of alerts for a managed target or a specific metric ordered by time.

Determine the notification status for a specific alert or set of alerts.

Generate statistical reports on alert frequency using criteria such as by target, by metric, by groups of metrics.

Custom implementation of event filtering or causal analysis.

Custom aggregation (grouping) of alert information into categories.

Custom alert status queries or aggregations for specific time periods. i.e. What was my most problematic metric for target x last week? How many times did alert y occur between times w and x?

Generating time based histograms of alerts.



The view does not force any specific order on the rows returned. The assumption is that if the user wants to order the alert records that are returned in a specific way, they will add an order by clause to their select query.

Access to this view will use an index if the query references the member target name, target type, metric name, and metric column or a subset of these columns if they are included as listed above from left to right.

Queries that use the collection\_timestamp or that query using the alert\_state will also be indexed.

## 6.2.14 MGMT\$METRIC\_DETAILS

MGMT\$METRIC\_DETAILS displays a rolling 25 hour window of individual metric samples. These are the metric values for the most recent sample that has been loaded into the Management Repository plus any earlier samples that have not been aggregated into hourly statistics.

**Table 6–14 MGMT\$METRIC\_DETAILS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_TYPE	VARCHAR2(20)	A DECODE of the internal numeric type of the metric that is being defined. This column will contain one of the following values: Number String Table Raw External
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.  For example, if this table describing the MGMT\$TARGET_TYPE view was being defined as a table metric, Column Name, Data Type, and Description would be metric columns.
METRIC_LABEL	VARCHAR2(64)	An intuitive display name for the metric that is being defined.
COLUMN_LABEL	VARCHAR2(64)	For table metrics, the column label contains an intuitive display name for the metric column.
COLLECTION_TIMESTAMP	DATE	The date-time when the alert condition was detected by the Management Agent.

**Table 6–14 (Cont.) MGMT\$METRIC\_DETAILS**

Column	Datatype	Description
VALUE	VARCHAR2(4000)	Since current metric values can be a numeric or a string type, this column returns the value of the metric as a string. If the user of the view is restricting the query to numeric metric values, they can use the TO_NUMBER SQL function to return the values in numeric form.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded

**Usage Notes**

Show the individual values for a metric over time.

Identify time periods when abnormal samples for metric were collected.

Calculate the correlation coefficient between two or more metrics.

Provide metric values that are associated with an alert.

Queries using this view will use an index if the queries use the target name, the target type, metric name, metric column, and key value, or if they are based upon the collection\_timestamp.

**6.2.15 MGMT\$METRIC\_CURRENT**

MGMT\$METRIC\_CURRENT displays information on the most recent metric values that have been loaded into the Management Repository.

**Table 6–15 MGMT\$METRIC\_CURRENT**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.

**Table 6–15 (Cont.) MGMT\$METRIC\_CURRENT**

Column	Datatype	Description
METRIC_TYPE	VARCHAR2(20)	A DECODE of the internal numeric type of the metric that is being defined. This column will contain one of the following values: Number String Table Raw External
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.  For example, if this table describing the MGMT\$TARGET_TYPE view was being defined as a table metric, Column Name, Data Type, and Description would be metric columns.
METRIC_LABEL	VARCHAR2(64)	An intuitive display name for the metric that is being defined.
COLUMN_LABEL	VARCHAR2(64)	For table metrics, the column label contains an intuitive display name for the metric column.
COLLECTION_TIMESTAMP	DATE	The date-time when the alert condition was detected by the Management Agent.
VALUE	VARCHAR2(4000)	Since current metric values can be a numeric or a string type, this column returns the value of the metric as a string. If the user of the view is restricting the query to numeric metric values, they can use the TO_NUMBER SQL function to return the values in numeric form.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded

**Usage Notes**

Retrieve the most recent value for a metric that is stored in the Management Repository.

Retrieve the latest metrics for a target or metric for a specific time period.

Queries using this view will use an index if the queries use target name, the target type, metric name, metric column, and key value, or if they are based upon the collection\_timestamp.

## 6.2.16 MGMT\$METRIC\_HOURLY

MGMT\$METRIC\_HOURLY displays metric statistics information that have been aggregated from the individual metric samples into hourly time periods. For example, if a metric is collected every 15 minutes, the 1 hour rollup would aggregate the 4 samples into a single hourly value by averaging the 4 individual samples together. The current hour of statistics may not be immediately available from this view. The timeliness of the information provided from this view is dependent on when the query against the view was executed and when the hourly rollup table was last refreshed.

**Table 6–16 MGMT\$METRIC\_HOURLY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.  For example, if this table describing the MGMT\$TARGET_TYPE view was being defined as a table metric, Column Name, Data Type, and Description would be metric columns.
METRIC_LABEL	VARCHAR2(64)	An intuitive display name for the metric that is being defined.
COLUMN_LABEL	VARCHAR2(64)	For table metrics, the column label contains an intuitive display name for the metric column.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
ROLLUP_TIMESTAMP	DATE	The rollup timestamp identifies the start of the rollup period. For the one-hour rollups, samples that fall within the hourly boundaries from minute 00 through minute 59 inclusive will be combined. For example, samples from 12:00 AM through 12:59 AM would be combined into a single aggregated record with a rollup timestamp of "date" 12:00 AM.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Table 6–16 (Cont.) MGMT\$METRIC\_HOURLY**

Column	Datatype	Description
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period.
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.

**Usage Notes**

This view provides the best level of granularity to show changes in a metric's value over the course of a day.

Identify hourly time periods when a metric or sets of metrics are maximized.

Understand how the variability of a metric over a one hour time period.

Identify the values of the collected metrics for a target when a particular hour has been identified as problematic.

Queries using this view will use an index if the queries use the target\_name, the metric\_name, or if they are based upon the rollup\_timestamp.

**6.2.17 MGMT\$METRIC\_DAILY**

MGMT\$METRIC\_DAILY displays metric statistics that have been aggregated from the samples collected over the previous twenty-four hour time period. The timeliness of the information provided from this view is dependent on when the query against the view was executed and when the hourly rollup table was last refreshed.

**Table 6–17 MGMT\$METRIC\_DAILY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.  For example, if this table describing the MGMT\$TARGET_TYPE view was being defined as a table metric, Column Name, Data Type, and Description would be metric columns.
METRIC_LABEL	VARCHAR2(64)	An intuitive display name for the metric that is being defined.

**Table 6–17 (Cont.) MGMT\$METRIC\_DAILY**

Column	Datatype	Description
COLUMN_LABEL	VARCHAR2(64)	For table metrics, the column label contains an intuitive display name for the metric column.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
ROLLUP_TIMESTAMP	DATE	The rollup timestamp identifies the start of the rollup period. For the one-hour rollups, samples that fall within the hourly boundaries from minute 00 through minute 59 inclusive will be combined. For example, samples from 12:00 AM through 12:59 AM would be combined into a single aggregated record with a rollup timestamp of "date" 12:00 AM.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period.
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.

**Usage Notes**

View provides the best granularity to show changes in a metric's value over the course of a week or month.

Understand trends in metric values.

Queries using this view will use an index if the queries use the target\_name, the metric\_name, or if they are based upon the rollup\_timestamp.

## 6.3 Inventory Views

### 6.3.1 MGMT\$TARGET

MGMT\$TARGET displays information about the managed targets that are known to the Management Repository. These targets may or may not be actively monitored.

**Table 6–18** *MGMT\$TARGET*

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The type of the target. Types of targets may include databases, hosts, web servers, applications, or Application Servers. The definer of the collection definition at the Management Agent defines the target type. The target type defines the set of metrics that are collected for a managed target within the Management Repository.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
TYPE_VERSION	VARCHAR2(8)	The target type meta version of the metadata set. Metadata versions may be updated when applying patches or upon new releases of Enterprise Manager Grid Control.
TYPE_QUALIFIER1-5	VARCHAR2(64)	Up to five qualifiers can be used to distinguish different metric definitions based on different system configurations. Example qualifier entries may include operating system version, database version, or RAC configuration.
EMD_URL	VARCHAR2(2000)	The URL address of the Management Agent that is managing the target.
TIMEZONE_REGION	VARCHAR2(64)	The timezone region in which the target operates.
DISPLAY_NAME	VARCHAR2(64)	User-friendly name for the target.
HOST_NAME	VARCHAR2(128)	Name of the host where the target is running. For composite targets or targets that span a host, this column will be NULL.
LAST_METRIC_LOAD_TIME	TIMESTAMP	Timestamp when information for this target was last loaded into the Management Repository. If metrics have not been loaded into the Management Repository for the target, this column will be NULL.
TYPE_DISPLAY_NAME	VARCHAR2(128)	User-friendly name of the target type.

**Usage Notes**

Display a list of the targets known to the Management Repository.

Display administration and monitoring information in the context of a managed target.

Order the targets by last load time for customers to get a sense on how recent the information is for a target in the Management Repository. To access this information in an ordered way, customers should use the appropriate ORDER BY clause with the view.

Access to this view will use an index if the query references the target name and target type.

There is an implicit assumption that customers will not use this view to identify the targets that are owned by a Management Agent or the targets that reside on a specific host.

## 6.3.2 MGMT\$TARGET\_TYPE

MGMT\$TARGET\_TYPE displays metric descriptions for a given target name and target type. This information is available for the metrics for the managed targets that have been loaded into the Management Repository. Metrics are specific to the target type.

**Table 6–19 MGMT\$TARGET\_TYPE**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
TYPE_VERSION	VARCHAR2(8)	The target type meta version of the metadata set. Metadata versions may be updated when applying patches or upon new releases of Enterprise Manager Grid Control.
TYPE_QUALIFIER1-5	VARCHAR2(64)	Up to five qualifiers can be used to distinguish different metric definitions based on different system configurations. Example qualifier entries may include operating system version, database version, or RAC configuration.
METRIC_NAME	VARCHAR2(64)	The name of the metric that is being defined.
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.  For example, if a table describing the MGMT\$TARGET_TYPE view is to be defined as a table metric, Column Name, Data Type, and Description would be metric columns.
KEY_COLUMN	VARCHAR2(64)	For table metrics, the key column contains the name of the column in the table that represents the primary key. Values in this column must uniquely identify rows in the table. If the metric that is being defined is not a table metric, the value in this column is a single space;  For example, the Column Name would be the key column if this table describing the MGMT\$TARGET_TYPE view was being defined as a table metric.
METRIC_TYPE	VARCHAR2(20)	A DECODE of the internal numeric type of the metric that is being defined. This column will contain one of the following values:  Number String Table Raw External Repository Metric
METRIC_LABEL	VARCHAR2(64)	A intuitive display name for the metric that is being defined.



**Table 6–19 (Cont.) MGMT\$TARGET\_TYPE**

Column	Datatype	Description
COLUMN_LABEL	VARCHAR2(64)	For table metrics, the column label contains a user understandable display name for the metric column.
DESCRIPTION	VARCHAR2(128)	A description of the metric that is being defined.
DESCRIPTION_NLSID	VARCHAR2(64)	The NLSid of the description of the metric
UNIT	VARCHAR2(32)	The unit of the metric that is being defined.
UNIT_NLSID	VARCHAR2(64)	The NLSid of the unit of the metric being defined.
SHORT_NAME	VARCHAR2(40)	This is a shortened version of the metric display name for the "dense" UI concept
SHORT_NAME_NLSID	VARCHAR2(64)	The NLSid of the short name of the metric being defined.

**Usage Notes**

List the set of metrics that have been defined for a target type.

Display intuitive metric names and associated attributes such as unit in a general way during portal, web application, or custom 4GL report generation.

Access to this view will use an index if the query references the metric name, metric column. The query should also qualify the target name and target type in order to restrict the amount of information returned.

**6.3.3 MGMT\$TARGET\_TYPE\_DEF**

MGMT\$TARGET\_TYPE\_DEF displays definition information for a target type.

**Table 6–20 MGMT\$TARGET\_TYPE\_DEF**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TYPE_DISPLAY_NAME	VARCHAR2(128)	User-friendly name of the target type.
TARGET_TYPE_GUID	RAW(16)	The unique global identifier (GUID) of the target type .
MAX_TYPE_META_VER	VARCHAR2(8)	The maximum version of the target type stored in the Management Repository.

**6.3.4 MGMT\$TARGET\_ASSOCIATIONS**

MGMT\$TARGET\_ASSOCIATIONS displays the various associations between targets. This view can be used to find all types of associations for a given target.

**Table 6–21 MGMT\$TARGET\_ASSOCIATIONS**

Column	Datatype	Description
ASSOC_DEF_NAME	VARCHAR2(64)	Name of the association definition.
SOURCE_TARGET_NAME	VARCHAR2(256)	Target name of the target to which the association is being defined.
SOURCE_TARGET_TYPE	VARCHAR2(64)	The target type of the target for which the association is being defined. "ANY" can be used to specify that any target type be used.

**Table 6–21 (Cont.) MGMT\$TARGET\_ASSOCIATIONS**

Column	Datatype	Description
ASSOC_TARGET_NAME	VARCHAR2(256)	Target Name of the target which is being associated with the source target.
ASSOC_TARGET_TYPE	VARCHAR2(64)	The target type of the associated target. "ANY" can be used to specify that any target type be used.
SCOPE_TARGET_NAME	VARCHAR2(256)	The target under whose scope the association is valid.  This applies to non-global associations only. For example: A database may be part of a composite target only for a particular service.
SCOPE_TARGET_TYPE	VARCHAR2(64)	The target type for which the association is valid. This applies to non-global associations only.
ASSOCIATION_TYPE	VARCHAR2(64)	The type of association.

**Usage Notes**

Can be used to list the associations defined for a specific target.

Queries using this view will use an index if either (source\_target\_name, source\_target\_type) or (assoc\_target\_name, assoc\_target\_type) is specified.

**6.3.5 MGMT\$TARGET\_MEMBERS**

MGMT\$TARGET\_MEMBERS displays the list of direct members for a target.

**Table 6–22 MGMT\$TARGET\_MEMBERS**

Column	Datatype	Description
AGGREGATE_TARGET_NAME	VARCHAR2(256)	Target name of the aggregate target.
AGGREGATE_TARGET_TYPE	VARCHAR2(64)	Target type of the aggregate target.
AGGREGATE_TARGET_GUID	RAW(16)	Target GUID of the aggregate target.
MEMBER_TARGET_NAME	VARCHAR2(256)	Target Name of the member target.
MEMBER_TARGET_TYPE	VARCHAR2(64)	Target type of the member target.
MEMBER_TARGET_GUID	RAW(16)	Target GUID of the member target.

**Usage Notes**

Find the members for a aggregate target.

Find the aggregate targets for which a given target is a direct member.

Queries, which specify values for (AGGREGATE\_TARGET\_NAME, AGGREGATE\_TARGET\_TYPE) or (MEMBER\_TARGET\_NAME, MEMBER\_TARGET\_TYPE) will use index.

Joins using AGGREGATE\_TARGET\_GUID and MEMBER\_TARGET\_GUID will be efficient.

**6.3.6 MGMT\$TARGET\_FLAT\_MEMBERS**

MGMT\$TARGET\_FLAT\_MEMBERS displays the list of all direct and indirect members of the target.

**Table 6–23 MGMT\$TARGET\_FLAT\_MEMBERS**

Column	Datatype	Description
AGGREGATE_TARGET_NAME	VARCHAR2(256)	The target name of the aggregate target
AGGREGATE_TARGET_TYPE	VARCHAR2(64)	The target type of the aggregate target
AGGREGATE_TARGET_GUID	RAW(16)	Target GUID of the aggregate target
MEMBER_TARGET_NAME	VARCHAR2(256)	Target Name of the member target
MEMBER_TARGET_TYPE	VARCHAR2(64)	Target type of the member target
MEMBER_TARGET_GUID	RAW(16)	Target GUID of the member target

**Usage Notes**

Find the members for a aggregate target.

Find the aggregate targets for which a given target is a member either directly or indirectly.

Queries, which specify values for (AGGREGATE\_TARGET\_NAME, AGGREGATE\_TARGET\_TYPE) or (MEMBER\_TARGET\_NAME, MEMBER\_TARGET\_TYPE) will use index.

Joins using AGGREGATE\_TARGET\_GUID and MEMBER\_TARGET\_GUID will be the most efficient on this view.

**6.3.7 MGMT\$TARGET\_TYPE\_PROPERTIES**

MGMT\$TARGET\_TYPE\_PROPERTIES displays the default list of properties that are applicable to the target based on the target type to which the target belongs.

**Table 6–24 MGMT\$TARGET\_TYPE\_PROPERTIES**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target.
TARGET_TYPE	VARCHAR2(64)	Name of the target type.
PROPERTY_NAME	VARCHAR2(64)	Name of the property, Example:is_aggregate,is_service,IsBaselineable.
PROPERTY_VALUE	VARCHAR2(1024)	Value of the property

**Usage Notes**

List the properties applicable to the target and the default values.

**6.3.8 MGMT\$TARGET\_PROPERTIES**

MGMT\$TARGET\_PROPERTIES displays detailed target properties.

**Table 6–25 MGMT\$**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.

**Table 6–25 (Cont.) MGMT\$**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
PROPERTY_NAME	VARCHAR2(64)	The name of the target property being defined.
PROPERTY_VALUE	VARCHAR2(1024)	The value of the target property being defined.
PROPERTY_TYPE	VARCHAR2(64)	The type of the target property being defined. Possible values are: INSTANCE, if the property is applicable to the target instance. DYNAMIC, if the property is calculated dynamically.

### 6.3.9 MGMT\$METRIC\_CATEGORIES

MGMT\$METRIC\_CATEGORIES displays the list of classes and categories to which the metric belongs. This view can be used to classify the metric based on the class (Example: Service, Functional) and category within the class (Example: security/configuration under functional class or usage/performance under service class).

**Table 6–26 MGMT\$METRIC\_CATEGORIES**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	Name of the target type.
TYPE_VERSION	VARCHAR2(8)	Version of the target type.
METRIC_NAME	VARCHAR2(64)	Name of the metric.
METRIC_COLUMN	VARCHAR2(64)	For table metrics, the metric column contains the name of the column in the table that is being defined. If the metric that is being defined is not a table metric, the value in this column is a single space.
METRIC_CLASS_NAME	VARCHAR2(64)	Name of the metric class to which this metric belongs.
METRIC_CATEGORY_NAME	VARCHAR2(64)	Name of the category of the metric in the class.
METRIC_CATEGORY_NLSID	VARCHAR2(64)	The NLS ID of the category which is used by Enterprise Manager to translate the name to different languages.

#### Usage Notes

Classify the metrics into different buckets based on Class and category.

## 6.4 Policy Definition Views

### 6.4.1 MGMT\$POLICIES

MGMT\$POLICIES displays the list of policies defined in the Management Repository.

**Table 6–27 MGMT\$POLICIES**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target.
METRIC_NAME	VARCHAR2(64)	The name of the metric based on which the policy is defined.
POLICY_NAME	VARCHAR2(128)	The name of the policy being defined.
POLICY_LABEL_NLSID	VARCHAR2(64)	The NSL ID for the policy name.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
AUTHOR	VARCHAR2(256)	The author of the policy.
DESCRIPTION	VARCHAR2(256)	The description of the policy.
DESCRIPTION_NLSID	VARCHAR2(64)	The NLS ID of the description.
IMPACT	VARCHAR2(500)	A description of the impact of the policy being defined.
IMPACT_NLSID	VARCHAR2(64)	NLSid for the policy impact.
RECOMMENDATION	VARCHAR2(500)	A recommendation for the policy being defined.
RECOMMENDATION_NLSID	VARCHAR2(64)	NLS ID for the policy recommendation.
VIOLATION_LEVEL	VARCHAR2(13)	The level at which the violations will be logged when the policy being defined is violated. Possible values are : Informational Warning Critical
CONDITION_TYPE	NUMBER(1)	The type condition used by the the policy being defined. Possible values are : 1. Simple condition 2. SQL expression condition 3. PL/SQL procedure for evaluating condition.
CONDITION	VARCHAR2(4000)	The condition details / expression used by the policy being defined. If the condition type is 1, this column contains the metric_column_name 2, this column contains the conditional expression 3, this column contains the pl/sql procedure name that performs the condition.
CONDITION_OPERATOR	NUMBER	Defines the policy condition operator to be applied, when the condition_type is 'Simple Condition'. Possible values are : 0 - GT 1 - EQ 2 - LT 3 - LE 4 - GE 5 - CONTAINS 6 - NE 7 - MATCH : regular expression
OWNER	VARCHAR2(256)	The Enterprise Manager administrator who created the policy in the repository.

**Table 6–27 (Cont.) MGMT\$POLICIES**

Column	Datatype	Description
AUTO_ENABLED	NUMBER(1)	Specifies whether the policy should be enable automatically on the new targets that are being added.
CATEGORY	VARCHAR2(64)	The name of the category the policy. Refer to MGMT\$METRIC_CATEGORIES for the list of all categories.
CATEGORY_NLSID	VARCHAR2(64)	The NLSid of the policy category.

## 6.4.2 MGMT\$POLICY\_PARAMETERS

MGMT\$POLICY\_PARAMETERS displays detailed information about a policy.

**Table 6–28 MGMT\$POLICY\_PARAMETERS**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and polcies that are applicable for the target.
METRIC_NAME	VARCHAR2(64)	The name of the metric based on which the policy is defined.
POLICY_NAME	VARCHAR2(128)	The name of the policy being defined.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
PARAMETER_NAME	VARCHAR2(64)	The name of the policy parameter
PARAMETER_NAME_NLSID	VARCHAR2(64)	The NLSid of the policy parameter.
PARAMETER_TYPE	VARCHAR2(7)	The type of the policy parameter. Possible values are : 1 for NUMERIC parameter 2 for STRING parameter

## 6.4.3 MGMT\$POLICY\_VIOLATION\_CTXT

MGMT\$POLICY\_VIOLATION\_CTXT displays the policy violation context.

**Table 6–29 MGMT\$POLICY\_VIOLATION\_CTXT**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and polcies that are applicable for the target
METRIC_NAME	VARCHAR2(64)	The name of the metric based on which the policy is defined.
POLICY_NAME	VARCHAR2(128)	The name of the policy being defined.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
COLUMN_NAME	VARCHAR2(64)	The column name to be included in the policy violation context.
IS_HIDDEN	NUMBER(1)	Flag indicating whether or not the value for this metric (data source) column should be displayed when viewing a violation. Default 0.
URL_LINK_TYPE	VARCHAR2(7)	Flag indicating the type of the link. Possible values : JSP UIX

**Table 6–29 (Cont.) MGMT\$POLICY\_VIOLATION\_CTXT**

Column	Datatype	Description
URL_LINK_TEMPLATE	VARCHAR2(4000)	URL template that contains references to any metric (data source) column name defined in the metric (data source). The actual values are substituted in at run-time to construct a URL that is associated with this metric columns value. Default NULL.

#### 6.4.4 MGMT\$TARGET\_POLICY\_EVAL\_SUMM

MGMT\$TARGET\_POLICY\_EVAL\_SUMM displays the policy violation summary for a target.

**Table 6–30 MGMT\$TARGET\_POLICY\_EVAL\_SUMM**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target for which the policies were being evaluated. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TARGET_GUID	RAW(16)	The unique global identifier for the target..
POLICY_NAME	VARCHAR2(128)	The name of the policy for which evaluation summary is provided.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
COLL_NAME	VARCHAR2(64)	The name of the collection on which the policy is evaluated.
LAST_EVALUATION_DATE	DATE	The last date on which the policy was evaluated for the target for the collection item.
TOTAL_VIOLATIONS	NUMBER	The number of violations logged for the given target policy association
NON_SUPPRESS_VIOLATIONS	NUMBER	The number of non-exempt violations logged for the given target policy association.
COMPLIANCE_SCORE	NUMBER	The compliance score calculated for the target policy association.

#### 6.4.5 MGMT\$POLICY\_VIOL\_ANNOTATIONS

MGMT\$POLICY\_VIOL\_ANNOTATIONS displays a summary of policy violation annotations.

**Table 6–31 MGMT\$POLICY\_VIOL\_ANNOTATIONS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.

**Table 6–31 (Cont.) MGMT\$POLICY\_VIOL\_ANNOTATIONS**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The unique global identifier for the target.
POLICY_NAME	VARCHAR2(128)	The name of the policy rule
DESCRIPTION	VARCHAR2(256)	A description of the policy rule
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
MESSAGE	VARCHAR2(4000)	The policy violation message.
VIOLATION_LEVEL	VARCHAR2(40)	A user readable description of the internal violation code that is sent from the Management Agent to identify the state of the policy violation. A policy violation record is transferred to the repository from the Management Agent each time the metric threshold is crossed in either direction, or if the Management Agent is restarted. The value of this column will contain one of the following strings: · Clear · Critical
COLLECTION_TIMESTAMP	DATE	The date-time when the policy violation was detected by the Management Agent.
ANNOTATION_MESSAGE	VARCHAR2(4000)	The annotation text.
ANNOTATION_TIMESTAMP	DATE	The time the annotation was created.
ANNOTATED_BY	VARCHAR2(256)	Enterprise Manager administrator who authored the the annotation.

## 6.4.6 MGMT\$POLICY\_VIOL\_NOTIF\_LOG

MGMT\$POLICY\_VIOL\_NOTIF\_LOG displays details of notification deliveries for policy violations.

**Table 6–32 MGMT\$POLICY\_VIOL\_NOTIF\_LOG**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target where the metric was collected. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
POLICY_NAME	VARCHAR2(128)	The name of the policy rule



**Table 6–32 (Cont.) MGMT\$POLICY\_VIOL\_NOTIF\_LOG**

Column	Datatype	Description
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
VIOLATION_MESSAGE	VARCHAR2(4000)	A user readable description of the internal violation code that is sent from the Management Agent to identify the state of the policy violation. A policy violation record is transferred to the repository from the Management Agent each time the metric threshold is crossed in either direction, or if the Management Agent is restarted. The value of this column will contain one of the following strings: <ul style="list-style-type: none"> <li>· Clear</li> <li>· Critical</li> </ul>
VIOLATION_LEVEL	VARCHAR2(40)	A user readable description of the internal violation code that is sent from the Management Agent to identify the state of the policy violation. A policy violation record is transferred to the repository from the Management Agent each time the metric threshold is crossed in either direction, or if the Management Agent is restarted. The value of this column will contain one of the following strings: <ul style="list-style-type: none"> <li>· Clear</li> <li>· Critical</li> </ul>
COLLECTION_TIMESTAMP	DATE	The date-time when the policy violation was detected by the Management Agent.
DELIVERY_MESSAGE	VARCHAR2(1024)	The message indicating the success or failure of the notification delivery.
DELIVERY_TIMESTAMP	DATE	The date-time when the notification was delivered.

## 6.5 Policy Association Views

### 6.5.1 MGMT\$TARGET\_POLICIES

MGMT\$TARGET\_POLICIES displays target policy information.

**Table 6–33** *MGMT\$TARGET\_POLICIES*

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target for which the policies were being evaluated. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TARGET_GUID	RAW(16)	The unique global identifier for the target.
POLICY_NAME	VARCHAR2(128)	The name of the policy that is associated with the target.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
CATEGORY	VARCHAR2(64)	The name of the category the policy. Refer to MGMT\$METRIC_CATEGORIES for the list of all categories.
IS_ENABLED	NUMBER(1)	Flag to indicate whether the policy is enabled or disabled on the target.

## 6.5.2 MGMT\$TARGET\_POLICY\_SETTINGS

MGMT\$TARGET\_POLICY\_SETTINGS displays policy settings for a target.

**Table 6–34** *MGMT\$TARGET\_POLICY\_SETTINGS*

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target for which the policies were being evaluated. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TARGET_GUID	RAW(16)	The unique global identifier for the target.
POLICY_NAME	VARCHAR2(128)	The name of the policy that is associated with the target.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
CATEGORY	VARCHAR2(64)	The name of the category the policy. Refer to MGMT\$METRIC_CATEGORIES for the list of all categories.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded

**Table 6–34 (Cont.) MGMT\$TARGET\_POLICY\_SETTINGS**

Column	Datatype	Description
KEY_OPERATOR	NUMBER	Specifies whether the key_value columns have any SQL wildcards.  For single key column metrics, the value is 1 if the key_value has wildcard characters, 0 otherwise.  For metrics with multiple keys, a list of operators for all key columns will be stored here. For example, a metric with 3 keys (k1, k2, k3) where K1 and K2 use wildcards and K3 uses exact match, then 011 is stored in this column
PARAMETER_NAME	VARCHAR2(64)	The name of the policy parameter.
PREVENT_OVERRIDE	NUMBER(1)	The is a flag that specifies that the metric rows with this key_value has a template override flag. Once the template override flag is ON, any template application will not update the threshold/parameter values.
POLICY_THRESHOLD	VARCHAR2(256)	The threshold value for the policy parameter.
ACTION_TYPE	VARCHAR2(17)	The corrective action type configured. Possible values are :  No-Action : when no action is configured Corrective-Action : when a repository side corrective action is configured Agent-Fixit Job : when an agent side fix-it job is configured.
ACTION_JOB_TYPE	VARCHAR2(32)	Specifies the job type of the corrective action when ACTION_TYPE is "Corrective-Action".
ACTION_JOB_NAME	VARCHAR2(64)	Specifies the job name of the corrective action when ACTION_TYPE is "Corrective-Action".
ACTION_JOB_OWNER	VARCHAR2(256)	Specifies the job owner of the corrective action when ACTION_TYPR is "Corrective-Action".

## 6.6 Policy Violation Views

### 6.6.1 MGMT\$POLICY\_VIOLATION\_CURRENT

MGMT\$POLICY\_VIOLATION\_CURRENT displays current information for any policy violations that are logged in the Management Repository that are in a non-clear state. Only the most recent open violation in a non-clear status for a given policy will be provided through this view.

**Table 6–35 MGMT\$POLICY\_VIOLATION\_CURRENT**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target for which the policies were being evaluated. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TYPE_DISPLAY_NAME	VARCHAR2(128)	User-friendly name of the target type.

**Table 6–35 (Cont.) MGMT\$POLICY\_VIOLATION\_CURRENT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The unique global identifier for the target.
POLICY_NAME	VARCHAR2(128)	The name of the policy that is associated with the target.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
CATEGORY	VARCHAR2(64)	The name of the category the policy. Refer to MGMT\$METRIC_CATEGORIES for the list of all categories.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
COLLECTION_TIMESTAMP	DATE	The date-time when the policy violation condition was detected.
VIOLATION_LEVEL	NUMBER	A user readable description of the internal violation level logged to identify the state of the policy condition. A violation record is logged each time the policy condition is violated in either direction. The value of this column will contain one of the following strings: · Informational · Warning · Critical If the policy condition goes into a clear state, it will no longer be visible from this view.
MESSAGE	VARCHAR2(4000)	An optional message that is generated when the violation is logged that provides additional information about the policy condition.
MESSAGE_NLSID	VARCHAR2(64)	The NLSid of the message.
MESSAGE_PARAMS	VARCHAR2(4000)	Contains the URL encoded parameters separated by "&" to be used to format the alert message.
SUPPRESS_CODE	NUMBER	Indication of exemption status. Legal values: 0 - not exempt 1 - exempt indefinitely 2 - exempt until date specified in SUPPRESS_UNTIL 3 - exempt until next evaluation Default 0'
SUPPRESS_UNTIL	DATE	Specifies the date until which exemption should be applied. Only application if SUPPRESS_CODE is 2
SUPPRESS_BY	VARCHAR2(256)	Enterprise Manager administrator that suppressed the violation.

## 6.6.2 MGMT\$POLICY\_VIOLATION\_HISTORY

MGMT\$POLICY\_VIOLATION\_HISTORY displays historical information for any policies that are logged in the Management Repository.

**Table 6–36 MGMT\$POLICY\_VIOLATION\_HISTORY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target for which the policies were being evaluated. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TYPE_DISPLAY_NAME	VARCHAR2(128)	User-friendly name of the target type.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
POLICY_NAME	VARCHAR2(128)	The name of the policy that is associated with the target.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
CATEGORY	VARCHAR2(64)	The name of the category the policy. Refer to MGMT\$METRIC_CATEGORIES for the list of all categories.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
COLLECTION_TIMESTAMP	DATE	The date-time when the policy violation condition was detected.
VIOLATION_LEVEL	NUMBER	A user readable description of the internal violation level logged to identify the state of the policy condition. A violation record is logged each time the policy condition is violated in either direction. The value of this column will contain one of the following strings: <ul style="list-style-type: none"> <li>· Informational</li> <li>· Warning</li> <li>· Critical</li> </ul> If the policy condition goes into a clear state, it will no longer be visible from this view.
VIOLATION_DURATION	NUMBER	The time, in hours, from when the policy condition was first detected until it was cleared.
MESSAGE	VARCHAR2(4000)	An optional message that is generated when the violation is logged that provides additional information about the policy condition.
MESSAGE_NLSID	VARCHAR2(64)	NLSid of the policy message.

**Table 6–36 (Cont.) MGMT\$POLICY\_VIOLATION\_HISTORY**

Column	Datatype	Description
MESSAGE_PARAMS	VARCHAR2(4000)	Contains the URL encoded parameters separated by "&" to be used to format the policy message.

### 6.6.3 MGMT\$POLICY\_VIOLATION\_CONTEXT

MGMT\$POLICY\_VIOLATION\_CONTEXT displays policy violation context.

**Table 6–37 MGMT\$POLICY\_VIOLATION\_CONTEXT**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Name of the target for which the policies were being evaluated. The target name uniquely identifies a managed target within the Management Repository. The target name typically contains the name of the managed entity that was provided by the system or database administrator.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TYPE_DISPLAY_NAME	VARCHAR2(128)	User-friendly name of the target type.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
POLICY_NAME	VARCHAR2(128)	The name of the policy that is associated with the target.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
CATEGORY	VARCHAR2(64)	The name of the category the policy. Refer to MGMT\$METRIC_CATEGORIES for the list of all categories.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
COLLECTION_TIMESTAMP	DATE	The date-time when the policy violation condition was detected.
COLUMN_NAME	VARCHAR2(64)	The associated name stored along with the policy violation.
COLUMN_VALUE	VARCHAR2(4000)	The associated value stored along with the policy violation.

## 6.7 Management Template Views

## 6.7.1 MGMT\$TEMPLATES

MGMT\$TEMPLATES displays details of all the management templates stored in the Management Repository.

**Table 6–38 MGMT\$TEMPLATES**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TEMPLATE_NAME	VARCHAR2(64)	The name of the template.
TEMPLATE_GUID	RAW(16)	The unique global identifier for the template.
DESCRIPTION	VARCHAR2(1024)	The description of the template.
OWNER	VARCHAR2(256)	Enterprise Manager administrator who owns the template.
IS_PUBLIC	NUMBER(1)	The flag to specify whether the template is accessible to all EM administrators.
CREATED_DATE	DATE	The date/time when the template is created in the repository.
LAST_UPDATED_DATE	DATE	The date/time when the template was last modified in the repository.
LAST_UPDATED_BY	VARCHAR2(256)	The EM administrator who last updated the template.

## 6.7.2 MGMT\$TEMPLATE\_POLICY\_SETTINGS

MGMT\$TEMPLATE\_POLICY\_SETTINGS displays policy settings for management templates.

**Table 6–39 MGMT\$TEMPLATE\_POLICY\_SETTINGS**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TEMPLATE_NAME	VARCHAR2(64)	The name of the template.
TEMPLATE_GUID	RAW(16)	The unique global identifier for the template.
POLICY_NAME	VARCHAR2(128)	The name of the policy that is associated with the template.
POLICY_GUID	RAW(16)	The unique global identifier for the policy.
CATEGORY	VARCHAR2(64)	The name of the category the policy. Refer to MGMT\$METRIC_CATEGORIES for the list of all categories.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded

**Table 6–39 (Cont.) MGMT\$TEMPLATE\_POLICY\_SETTINGS**

Column	Datatype	Description
KEY_OPERATOR	NUMBER	Specifies whether the key_value columns have any SQL wildcards.  For single key column metrics, the value is 1 if the key_value has wildcard characters, 0 otherwise.  For metrics with multiple keys, a list of operators for all key columns will be stored here. For example, a metric with 3 keys (k1, k2, k3) where K1 and K2 use wildcards and K3 uses exact match, then 011 is stored in this column
PARAMETER_NAME	VARCHAR2(64)	The name of the parameter.
PREVENT_OVERRIDE	NUMBER(1)	The is a flag that specifies that the metric rows with this key_value has a tempalte override flag. Once the tempalte override flag is ON, any template application will not update the threshold/parameter values.
POLICY_THRESHOLD	VARCHAR2(256)	The threshold value confugured for the policy parameter.
ACTION_TYPE	VARCHAR2(17)	The corrective action type configured. Possible values are :  No-Action : when no action is configured Corrective-Action : when a repository side corrective action is configured Agent-Fixit Job : when an agent side fix-it job is configured.
ACTION_JOB_TYPE	VARCHAR2(32)	Specifies the job type of the corrective action when ACTION_TYPE is "Corrective-Action".
ACTION_JOB_NAME	VARCHAR2(64)	Specifies the job name of the corrective action when ACTION_TYPE is "Corrective-Action".
ACTION_JOB_OWNER	VARCHAR2(256)	Specifies the job owner of the corrective action when ACTION_TYPE is "Corrective-Action".

### 6.7.3 MGMT\$TEMPLATE\_METRICCOLLECTION

MGMT\$TEMPLATE\_METRICCOLLECTIONS displays information on the metric collections defined for a template.

**Table 6–40 MGMT\$TEMPLATE\_METRICCOLLECTION**

Column	Datatype	Description
TEMPLATE_NAME	VARCHAR2(64)	The name of the template.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TEMPLATE_GUID	RAW(16)	The unique global identifier for the template.
METRIC_NAME	VARCHAR2(64)	The name of the metric for which the template collection is configured.
METRIC_COLUMN	VARCHAR2(64)	The name of the metric column for which the template collection is configured.
METRIC_GUID	RAW(16)	The unique global identifier for the metric column.
COLLECTION_NAME	VARCHAR2(64)	The name of the collection.



**Table 6–40 (Cont.) MGMT\$TEMPLATE\_METRICCOLLECTION**

Column	Datatype	Description
IS_REPOSITORY	NUMBER(1)	Indicates whether this is a repository-side collection. A repository-side collection has a PL/SQL evaluation procedure that is responsible for calculating the metric values.
FREQUENCY_CODE	VARCHAR2(9)	The metric collection frequency type. Possible values are: 1: One Time 2: Interval 3: Daily 4: Weekly 5: Monthly 6: Yearly 7: On Demand
COLLECTION_FREQUENCY	VARCHAR2(81)	The frequency of the metric collection. Value displayed is dependent on the frequency code: For One Time, the start date-time is stored in DD-MON-YY HH24:MI format. For Interval type, the frequency in minutes is stored. For Daily/Weekly/Monthly/Yearly types, the hour and minute of collection is stored in HH24:MI format. For On-Demand type, On-Demand is stored.
UPLOAD_POLICY	NUMBER	The frequency with which the metric data is uploaded/stored.

## 6.7.4 MGMT\$TEMPLATE\_METRIC\_SETTINGS

MGMT\$TEMPLATE\_METRIC\_SETTINGS displays management template settings.

**Table 6–41 MGMT\$TEMPLATE\_METRIC\_SETTINGS**

Column	Datatype	Description
TEMPLATE_NAME	VARCHAR2(64)	The name of the template.
TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics and policies that are applicable for the target
TEMPLATE_GUID	RAW(16)	The unique global identifier for the template.
METRIC_NAME	VARCHAR2(64)	The name of the metric for which the template collection is configured.
METRIC_COLUMN	VARCHAR2(64)	The name of the metric column for which the template collection is configured.
METRIC_GUID	RAW(16)	The unique global identifier for the metric column.
COLLECTION_NAME	VARCHAR2(64)	The name of the collection.
CATEGORY	VARCHAR2(64)	The name of the category the policy. Refer to MGMT\$METRIC_CATEGORIES for the list of all categories.
KEY_VALUE	VARCHAR2(256)	The key value for which the alert has been recorded. For composite keys, this is the first part of the key.
KEY_VALUE2	VARCHAR2(256)	For composite keys, this is the second part of the key for which the alert has been recorded

**Table 6–41 (Cont.) MGMT\$TEMPLATE\_METRIC\_SETTINGS**

Column	Datatype	Description
KEY_VALUE3	VARCHAR2(256)	For composite keys, this is the third part of the key for which the alert has been recorded
KEY_VALUE4	VARCHAR2(256)	For composite keys, this is the fourth part of the key for which the alert has been recorded
KEY_VALUE5	VARCHAR2(256)	For composite keys, this is the fifth part of the key for which the alert has been recorded
KEY_OPERATOR	NUMBER	Specifies whether the key_value columns have any SQL wildcards.  For single key column metrics, the value is 1 if the key_value has wildcard characters, 0 otherwise.  For metrics with multiple keys, a list of operators for all key columns will be stored here. For example, a metric with 3 keys (k1, k2, k3) where K1 and K2 use wildcards and K3 uses exact match, then 011 is stored in this column
PREVENT_OVERRIDE	NUMBER(1)	The is a flag that specifies that the metric rows with this key_value has a template override flag. Once the template override flag is ON, any template application will not update the threshold/parameter values.
WARNING_OPERATOR	NUMBER	Defines the warning threshold condition to be applied 0 - GT 1 - EQ 2 - LT 3 - LE 4 - GE 5 - CONTAINS 6 - NE 7 - MATCH : regular expression
WARNING_THRESHOLD	VARCHAR2(256)	The value of the warning threshold.
CRITICAL_OPERATOR	NUMBER	Defines the critical threshold condition to be applied 0 - GT 1 - EQ 2 - LT 3 - LE 4 - GE 5 - CONTAINS 6 - NE 7 - MATCH : regular expression
CRITICAL_THRESHOLD	VARCHAR2(256)	The value of the critical threshold.
OCCURRENCE_COUNT	NUMBER	The number of times the test has to trigger to raise a violation.
WARNING_ACTION_TYPE	VARCHAR2(17)	The warning corrective action type configured. Possible values are : No-Action : when no action is configured Corrective-Action : when a repository side corrective action is configured Agent-Fixit Job : when an agent side fix-it job is configured.

**Table 6–41 (Cont.) MGMT\$TEMPLATE\_METRIC\_SETTINGS**

Column	Datatype	Description
WARNING_ACTION_JOB_TYPE	VARCHAR2(32)	Specifies the job type of the warning corrective action when WARNING_ACTION_TYPE is "Corrective-Action".
WARNING_ACTION_JOB_OWNER	VARCHAR2(256)	Specifies the job owner of the warning corrective action when WARNING_ACTION_TYPE is "Corrective-Action".
WARNING_ACTION_JOB_NAME	VARCHAR2(64)	Specifies the job name of the warning corrective action when WARNING_ACTION_TYPE is "Corrective-Action".
CRITICAL_ACTION_TYPE	VARCHAR2(17)	The critical corrective action type configured. Possible values are : No-Action : when no action is configured Corrective-Action : when a repository side corrective action is configured Agent-Fixit Job : when an
CRITICAL_ACTION_JOB_TYPE	VARCHAR2(32)	Specifies the job type of the critical corrective action when CRITICAL_ACTION_TYPE is "Corrective-Action".
CRITICAL_ACTION_JOB_OWNER	VARCHAR2(256)	Specifies the job owner of the critical corrective action when CRITICAL_ACTION_TYPE is "Corrective-Action".
CRITICAL_ACTION_JOB_NAME	VARCHAR2(64)	Specifies the job name of the critical corrective action when CRITICAL_ACTION_TYPE is "Corrective-Action".

## 6.8 Job Views

### 6.8.1 MGMT\$JOBS

MGMT\$JOBS displays information about a job including the job's schedule.

**Table 6–42 MGMT\$JOBS**

Column	Datatype	Description
JOB_NAME	VARCHAR2(64)	The unique name for the job.
JOB_ID	RAW(16)	The unique system identifier for the job.
JOB_OWNER	VARCHAR2(256)	The owner/creator of the job.
JOB_DESCRIPTION	VARCHAR2(4000)	Optional text describing the job function.
JOB_TYPE	VARCHAR2(32)	The job type. For example, multi-task, SQL script or OS Command.
TARGET_TYPE	VARCHAR2()	The type of target the job was submitted against. Applies to single-target jobs only.
IS_LIBRARY	NUMBER(1)	Indicates whether or not the job is part of the job library.
IS_RESTARTABLE	NUMBER(1)	Indicates whether the job can be restarted. "0" indicates the job is not restartable. "1" indicates the job is not restartable.  By default, a job is not restartable if the original job owner is deleted and the job is transferred to another administrator.

**Table 6–42 (Cont.) MGMT\$JOBS**

Column	Datatype	Description
START_TIME	DATE	The scheduled start time. For daily, days of week and days of month schedules, the start_time denotes when the job should start.
END_TIME	DATE	For all periodic schedules, the last date (and time) to run the job. For daily, day of week and day of month schedules, only the date portion is used.
TIMEZONE_TYPE		Possible values are 1 - Repository timezone 2 - Target timezone 4 - Specified timezone region
TIMEZONE_REGION	VARCHAR2(64)	The specified timezone region.
SCHEDULE_TYPE		Possible values are: 0 - Immediate schedule 1 - Run once at specified time 2 - Run on interval 3 - Run daily 4 - Run on specified days of the week 5 - Run on specified days of the month 6 - Run on specified days of the year
INTERVAL	NUMBER	If schedule_type is interval (2), this is the interval at which the job repeats, in minutes
EXECUTION_HOURS	NUMBER(3)	Indicates the time of day at which the job will execute. Hours are specified using the 24-hour format (0 to 23).
EXECUTION_MINUTES	NUMBER(3)	Indicates the time of day at which the job will execute. Minutes are specified as a number between 0 and 59.
MONTHS	Integer Array	For days-of-year job schedules, this indicates the "month" in the schedule.
DAYS	Integer Array	For day-of-week/month or day(s) of the week job schedules, this indicates the "day" of the week/month.  Days-of-week specified as numbers 1 (Sunday) to 7 (Saturday).  Days-of-month specified as numbers 1 to 31.

## 6.8.2 MGMT\$JOB\_TARGETS

MGMT\$JOB\_TARGETS displays the target(s) the job was submitted against.

**Table 6–43 MGMT\$JOB\_TARGETS**

Column	Datatype	Description
JOB_NAME	VARCHAR2(64)	The unique name for the job.
JOB_OWNER	VARCHAR2(256)	The owner/creator of the job.
JOB_TYPE	VARCHAR2(32)	The job type. For example, multi-task, SQL script or OS Command.
TARGET_NAME	VARCHAR2(64)	Name of the target the job was submitted against.
TARGET_TYPE	VARCHAR2()	The type of target the job was submitted against. Applies to single-target jobs only.

**Table 6–43 (Cont.) MGMT\$JOB\_TARGETS**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The unique global identifier for the target.

### 6.8.3 MGMT\$JOB\_EXECUTION\_HISTORY

MGMT\$JOB\_EXECUTION\_HISTORY displays a summary of job executions along with their status and targets for each execution.

**Table 6–44 MGMT\$JOB\_EXECUTION\_HISTORY**

Column	Datatype	Description
JOB_NAME	VARCHAR2(64)	The unique name for the job.
JOB_OWNER	VARCHAR2(256)	The owner/creator of the job.
JOB_ID	RAW(16)	The unique system identifier for the job.
JOB_TYPE	VARCHAR2(32)	The job type. For example, multi-task, SQL script or OS Command.
EXECUTION_ID	RAW(16)	The unique execution identifier.
SCHEDULED_TIME	DATE	The scheduled time of job execution.
START_TIME	DATE	The actual time the job executed.
END_TIME	DATE	The actual time the job terminated execution.
STATUS	INTEGER	The current status of the job execution represented by integers 1 to 18: 1: Scheduled 2: Running 3: Failed Initialization 4: Failed 5: Succeeded 6: Suspended by User 7: Suspended: Agent Unreachable 8: Stopped 9: Suspended on Lock 10: Suspended on Event 11: Stop Pending 13: Suspend Pending 14: Inactive 15: Queued 16: Failed Retried 18: Skipped
TARGET_NAME	VARCHAR2(64)	Name of the target the job was submitted against.
TARGET_TYPE	VARCHAR2()	The type of target the job was submitted against. Applies to single-target jobs only.
TARGET_GUID	RAW(16)	The unique global identifier for the target.

### 6.8.4 MGMT\$JOB\_STEP\_HISTORY

MGMT\$JOB\_STEP\_HISTORY displays step-level details of job executions.

**Table 6–45 MGMT\$JOB\_STEP\_HISTORY**

Column	Datatype	Description
JOB_NAME	VARCHAR2(64)	The unique name for the job.
JOB_OWNER	VARCHAR2(256)	The owner/creator of the job.
JOB_ID	RAW(16)	The unique system identifier for the job.
JOB_TYPE	VARCHAR2(32)	The job type. For example, multi-task, SQL script or OS Command.
EXECUTION_ID	RAW(16)	The unique execution identifier.
STEP_NAME	VARCHAR2(variable )	The name of the job step.
START_TIME	DATE	The start time of the job step.
END_TIME	DATE	The end time of the job step.
STATUS	INTEGER	The current status of the job execution represented by integers 1 to 18: 1: Scheduled 2: Running 3: Failed Initialization 4: Failed 5: Succeeded 6: Suspended by User 7: Suspended: Agent Unreachable 8: Stopped 9: Suspended on Lock 10: Suspended on Event 11: Stop Pending 13: Suspend Pending 14: Inactive 15: Queued 16: Failed Retried 18: Skipped
TARGET_NAME	VARCHAR2(64)	Name of the target the job was submitted against.
TARGET_TYPE	VARCHAR2()	The type of target the job was submitted against. Applies to single-target jobs only.
TARGET_GUID	RAW(16)	The unique global identifier for the target.
OUTPUT	CLOB	Generated job output.

## 6.8.5 MGMT\$JOB\_ANNOTATIONS

MGMT\$JOB\_ANNOTATIONS displays a summary of annotations for changes in job status.

**Table 6–46 MGMT\$JOB\_ANNOTATIONS**

Column	Datatype	Description
JOB_NAME	VARCHAR2(64)	The unique name for the job.
JOB_OWNER	VARCHAR2(256)	The owner/creator of the job.

**Table 6–46 (Cont.) MGMT\$JOB\_ANNOTATIONS**

Column	Datatype	Description
JOB_STATUS	NUMBER(2)	The job status. Possible values are as follows: 1: Scheduled 2: Executing 3: Aborted 4: Failed 5: Completed 6: Suspended 7: Agent Down 8: Stopped 9: Suspended/Lock 11: Suspended/Blackout 13: Suspend Pending 15: Queued 16: Failed 17: Waiting 18: Skipped
OCCURRENCE_TIMESTAMP	DATE	The time at which the state change occurred.
ANNOTATION_MESSAGE	VARCHAR2(4000)	Annotation text.
ANNOTATION_TIMESTAMP	DATE	The time the annotation was created.
ANNOTATED_BY	VARCHAR2(256)	Enterprise Manager administrator who authored the the annotation.

## 6.8.6 MGMT\$JOB\_NOTIFICATION\_LOG

MGMT\$JOB\_NOTIFICATION\_LOG displays details of notification deliveries for changes in job status.

**Table 6–47 MGMT\$JOB\_NOTIFICATION\_LOG**

Column	Datatype	Description
JOB_NAME	VARCHAR2(64)	The unique name for the job.
JOB_OWNER	VARCHAR2(256)	The owner/creator of the job.
JOB_STATUS	NUMBER(2)	The job status. Possible values are as follows: 1: Scheduled 2: Executing 3: Aborted 4: Failed 5: Completed 6: Suspended 7: Agent Down 8: Stopped 9: Suspended/Lock 11: Suspended/Blackout 13: Suspend Pending 15: Queued 16: Failed 17: Waiting 18: Skipped

**Table 6–47 (Cont.) MGMT\$JOB\_NOTIFICATION\_LOG**

Column	Datatype	Description
OCCURRENCE_TIMESTAMP	DATE	The time at which the state change occurred.
DELIVERY_MESSAGE	VARCHAR2(1024)	The message indicating the success or failure of the notification delivery.
DELIVERY_TIMESTAMP	DATE	The time at which the log message was created.

## 6.9 Application Service Level Management Views

### 6.9.1 MGMT\$CSM\_REGION

MGMT\$CSM\_REGION displays a list of regions and their member targets.

**Table 6–48 MGMT\$CSM\_REGION**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target from which the metrics will be collected (Webcache target).
TARGET_TYPE	VARCHAR2(64)	Type defining the set of metrics that are applicable for the target.
REGION_NAME	VARCHAR2(128)	A meaningful name for a logical group (region) containing one or more domains. For example, the REGION_NAME can be a country, company, or company division.
DESCRIPTION	VARCHAR2(256)	A useful description of region.
REGION_MEMBER	VARCHAR2(256)	A member of the region specified by REGION_NAME. REGION_MEMBER can be a domain name or an IP subnet address.
REGION_MEMBER_TYPE	VARCHAR2(16)	Specifies the format type (domain name or IP subnet address) of REGION_MEMBER. Member type can be specified using "D" for domain or "S" for subnet. For example: REGION_MEMBER=us.oracle MEMBER_TYPE=D REGION_MEMBER=138.1 MEMBER_TYPE=S

#### Usage Notes

List the regions with detailed information.

List the regions of which a target is a member.

List the targets that belong to a region

### 6.9.2 MGMT\$CSM\_WATCHLIST

MGMT\$CSM\_WATCHLIST displays a list of URLs defined as part of a website target's watchlist. These URLs are periodically visited to monitor the website.



**Table 6–49 MGMT\$CSM\_WATCHLIST**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target from which the metrics will be collected (Webcache target).
TARGET_TYPE	VARCHAR2(64)	Type defining the set of metrics that are applicable for the target.
DISPLAY_NAME	VARCHAR2(128)	Display name for the URL.
URL	VARCHAR2(512)	Full URL filename containing the relative path from the web server. (For example, "/somepath/somefile.html").
DESCRIPTION	VARCHAR2(256)	Text description of the URL. (Optional)

**Usage Notes**

List the URL watchlist for a given target

**6.9.3 MGMT\$CSM\_METRIC\_DETAILS**

MGMT\$CSM\_METRIC\_DETAILS displays detailed information on each individual HTTP request for composite targets (for example, a website target) that has one or more Web Cache targets.

**Table 6–50 MGMT\$CSM\_METRIC\_DETAILS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	Name of the target from which the metrics will be collected (Webcache target).
TARGET_TYPE	VARCHAR2(64)	Type defining the set of metrics that are applicable for the target.
COLLECTION_TIMESTAMP	DATE	The date/time when the metric was collected
METRIC_NAME	VARCHAR2(64)	The name of the metric that is being defined
METRIC_VALUE	NUMBER	Number of milliseconds required to access this URL. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
URL	VARCHAR2(1024)	URL filename relative to the webserver host and port. The response of this url is measured in elapsed_time.
VISITOR_IP	VARCHAR2(32)	IP address of machine making requests
VISITOR_NODE	VARCHAR2(1024)	Resolved machine name from which a logged request is made.
VISITOR_DOMAIN	VARCHAR2(1024)	Domain of the machine making requests.
VISITOR_SUBNET	VARCHAR2(32)	Domain subnet of the machine making the requests.

**Usage Notes**

Analyze HTTP request patterns over time with regards to visitor information (IP, subnet, domain, region) or URLs (most requested, least requested) for composite targets.

Analyze HTTP request response times over time with regards to visitor information (IP, subnet, domain, region) and URLs for composite targets.

## 6.9.4 MGMT\$CSM\_MT\_METRIC\_DETAILS

MGMT\$CSM\_MT\_METRIC\_DETAILS displays detailed information on each individual HTTP request for Web Cache targets (i.e. member targets). If a Web Cache target is not a member of any composite targets, the composite target columns are presented as NULL. If a Web Cache target is a member of multiple composite targets, a row for each composite target is presented.

**Table 6–51 MGMT\$CSM\_MT\_METRIC\_DETAILS**

Column	Datatype	Description
MEMBER_TARGET_NAME	VARCHAR2(64)	Name of the Web Cache target.
MEMBER_TARGET_TYPE	VARCHAR2(64)	Type of the Web Cache target.
COMPOSITE_TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.  This entry may be null if the Web Cache target is not a member of any composite targets.
COMPOSITE_TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the composite target.  This entry may be null if the Web Cache target is not a member of any composite targets.
COLLECTION_TIMESTAMP	DATE	The time when the target status change was last detected and logged in the Management Repository.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
URL	VARCHAR2(1024)	URL filename relative to the webserver host and port.
VISITOR_IP	VARCHAR2(32)	IP address of machine making requests
VISITOR_NODE	VARCHAR2(1024)	Resolved machine name from which a logged request is made.
VISITOR_DOMAIN	VARCHAR2(1024)	Domain of the machine making requests.
VISITOR_SUBNET	VARCHAR2(32)	Domain subnet of the machine making the requests.

### Usage Notes

Analyze HTTP requests patterns over time with regards to visitor information (IP, subnet, domain, region) or URLs (most requested, least requested) for a particular Web Cache member target of a composite target, or for Web Cache targets that are not a member of any composite targets.

Analyze HTTP request response times over time with regards to visitor information (IP, subnet, domain, region) and URLs for a particular Web Cache member target of a composite target, or for Web Cache targets that are not a member of any composite targets.

## 6.9.5 MGMT\$CSM\_URL\_HOURLY

MGMT\$CSM\_URL\_HOURLY displays statistical information about the HTTP requests for a given composite target, aggregated by URL in hourly time periods.

**Table 6–52 MGMT\$CSM\_URL\_HOURLY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.  This entry may be null if the Web Cache target is not a member of any composite targets.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the composite target. This entry may be null if the Web Cache target is not a member of any composite targets.
URL	VARCHAR2(1024)	URL filename relative to the webserver host and port.
ROLLUP_TIMESTAMP	DATE	Identifies the start of the rollup period. For the one-hour rollups, samples that fall within the hourly boundaries from minute 00 through minute 59 inclusive will be combined. For example, samples from 12:00 AM through 12:59 AM would be combined into a single aggregated record with a rollup timestamp of "date" 12:00 AM.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The average of the metric values for the samples that have been included in the rollup period
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze hourly HTTP response time patterns and statistical (min, max, avg, stddev) information for a given URL of a composite target. (For example, what was the average response time experienced when accessing URL HTTP://my.oracle.com/home.html of the website my.oracle.com between 8 and 9 this morning).

**6.9.6 MGMT\$CSM\_URL\_DAILY**

MGMT\$CSM\_URL\_DAILY displays statistical information about the HTTP requests for a given composite target, aggregated by URL in daily time periods.

**Table 6–53 MGMT\$CSM\_URL\_DAILY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.

**Table 6–53 (Cont.) MGMT\$CSM\_URL\_DAILY**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
URL	VARCHAR2(1024)	Full URL filename containing the relative path from the web server. (For example, "/somepath/somefile.html").
ROLLUP_TIMESTAMP	DATE	Identifies the start of the rollup period. For the one-hour rollups, samples that fall within the hourly boundaries from minute 00 through minute 59 inclusive will be combined. For example, samples from 12:00 AM through 12:59 AM would be combined into a single aggregated record with a rollup timestamp of "date" 12:00 AM.
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The average of the metric values for the samples that have been included in the rollup period
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze daily HTTP response time patterns and statistical (min, max, avg, stddev) information for a given URL of a composite target. (For example, what was the avg response time experienced when accessing URL HTTP://my.oracle.com/home.html of the website my.oracle.com last Monday).

**6.9.7 MGMT\$CSM\_URL\_DIST\_HOURLY**

MGMT\$CSM\_URL\_DIST\_HOURLY displays the distribution of the HTTP request response times, in seconds, for a URL of a composite target. The data is aggregated in hourly time periods.

**Table 6–54 MGMT\$CSM\_URL\_DIST\_HOURLY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
URL	VARCHAR2(1024)	Full URL filename containing the relative path from the web server. (For example, "/somepath/somefile.html").

**Table 6–54 (Cont.) MGMT\$CSM\_URL\_DIST\_HOURLY**

Column	Datatype	Description
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Usage Notes**

Analyze the distribution of the HTTP response times for a URL of a composite target during a specific hour of the day (For example, compare the percentage of requests which response time was longer than 5 seconds at 8am and at 5pm).

**6.9.8 MGMT\$CSM\_URL\_DIST\_DAILY**

MGMT\$CSM\_URL\_DIST\_DAILY displays the distribution of the HTTP request response times, in seconds, for a URL of a composite target. The data is aggregated in daily time periods.

**Table 6–55 MGMT\$CSM\_URL\_DIST\_DAILY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
URL	VARCHAR2(1024)	Full URL filename containing the relative path from the web server. (For example, "/somepath/somefile.html").
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Usage Notes**

Analyze the distribution of the HTTP response times for a given URL of a composite target during a specific day (For example, compare the percentage of requests which response time was longer than 5 seconds on Monday and on Saturday).

**6.9.9 MGMT\$CSM\_MT\_URL\_HOURLY**

MGMT\$CSM\_MT\_URL\_HOURLY displays statistical information about the HTTP requests for a given Web Cache target of a composite target, aggregated by URL in hourly time periods. If a Web Cache target is not a member of any composite targets,

the composite target columns are presented as NULL. If a Web Cache target is a member of multiple composite targets, a row for each composite target is presented.

**Table 6–56 MGMT\$CSM\_MT\_URL\_HOURLY**

Column	Datatype	Description
MEMBER_TARGET_NAME	VARCHAR2(64)	The Web Cache target.
MEMBER_TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
COMPOSITE_TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table. May be null if the Web Cache target is not a member of any composite targets.
COMPOSITE_TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target. May be null if the Web Cache target is not a member of any composite target.
URL	VARCHAR2(1024)	Full URL filename containing the relative path from the web server. (For example, "/somepath/somefile.html").
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

### Usage Notes

Analyze hourly HTTP response time patterns and statistical (min, max, avg, stddev) information for a given URL of a Web Cache target and optionally of a composite target (For example, what was the avg response time experienced when accessing URL HTTP://my.oracle.com/home.html at Web Cache WEB5 of the website my.oracle.com between 8 and 9 this morning).

### 6.9.10 MGMT\$CSM\_MT\_URL\_DAILY

MGMT\$CSM\_MT\_URL\_DAILY displays statistical information about the HTTP requests for a given Web Cache target of a composite target, aggregated by URL in daily time periods. If a Web Cache target is not a member of any composite targets, the composite target columns are presented as NULL. If a Web Cache target is a member of multiple composite targets, a row for each composite target is presented.

**Table 6–57** *MGMT\$CSM\_MT\_URL\_DAILY*

Column	Datatype	Description
MEMBER_TARGET_NAME	VARCHAR2(64)	The Web Cache target.
MEMBER_TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
COMPOSITE_TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table. May be null if the Web Cache target is not a member of any composite targets.
COMPOSITE_TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target. May be null if the Web Cache target is not a member of any composite targets.
URL	VARCHAR2(1024)	Full URL filename containing the relative path from the web server. (For example, "/somepath/somefile.html").
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze daily HTTP response time patterns and statistical (min, max, avg, stddev) information for a given URL of a Web Cache target and optionally of a composite target. target (For example, what was the avg response time experienced when accessing URL HTTP://my.oracle.com/home.html at Web Cache WEB5 of the website my.oracle.com last Monday).

**6.9.11** *MGMT\$CSM\_MT\_URL\_DIST\_HOURLY*

*MGMT\$CSM\_MT\_URL\_DIST\_HOURLY* displays the distribution of the HTTP request response times, in seconds, for a URL of a given Web Cache target of a composite target. The data is aggregated in hourly time periods. If a Web Cache target is not a member of any composite targets, the composite target columns are presented as NULL. If a Web Cache target is a member of multiple composite targets, a row for each composite target is presented.

**Table 6–58** *MGMT\$CSM\_MT\_URL\_DIST\_HOURLY*

Column	Datatype	Description
MEMBER_TARGET_NAME	VARCHAR2(64)	The Web Cache target.

**Table 6–58 (Cont.) MGMT\$CSM\_MT\_URL\_DIST\_HOURLY**

Column	Datatype	Description
MEMBER_TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
COMPOSITE_TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table. May be null if the Web Cache target is not a member of any composite targets.
COMPOSITE_TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target. May be null if the Web Cache target is not a member of any composite targets.
URL	VARCHAR2(1024)	Full URL filename containing the relative path from the web server. (For example, "/somepath/somefile.html").
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Usage Notes**

Analyze the distribution of the HTTP response times for a URL of a Web Cache target of a composite target during a specific hour of the day (For example, how many requests for URL HTTP://my.oracle.com/home.html at Web Cache WEB5 of the website my.oracle.com took longer than 5 seconds between 8 and 9 this morning).

**6.9.12 MGMT\$CSM\_MT\_URL\_DIST\_DAILY**

MGMT\$CSM\_MT\_URL\_DIST\_DAILY displays the distribution of the HTTP request response times, in seconds, for a URL of a given Web Cache target of a composite target. The data is aggregated in daily time periods. If a Web Cache target is not a member of any composite targets, the composite target columns are presented as NULL. If a Web Cache target is a member of multiple composite targets, a row for each composite target is presented.

**Table 6–59 MGMT\$CSM\_MT\_URL\_DIST\_DAILY**

Column	Datatype	Description
MEMBER_TARGET_NAME	VARCHAR2(64)	The Web Cache target.
MEMBER_TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
COMPOSITE_TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table. May be null if the Web Cache target is not a member of any composite targets.
COMPOSITE_TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target. May be null if the Web Cache target is not a member of any composite targets.



**Table 6–59 (Cont.) MGMT\$CSM\_MT\_URL\_DIST\_DAILY**

Column	Datatype	Description
URL	VARCHAR2(1024)	Full URL filename containing the relative path from the web server. (For example, "/somepath/somefile.html").
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Usage Notes**

Analyze the distribution of the HTTP response times for a URL of a Web Cache target of a composite target during a specific day (For example, how many requests for URL HTTP://my.oracle.com/home.html at Web Cache WEB5 of the website my.oracle.com took longer than 5 seconds last Monday).

**6.9.13 MGMT\$CSM\_IP\_HOURLY**

MGMT\$CSM\_IP\_HOURLY displays statistical information about the HTTP requests for a given composite target, aggregated by the visitor IP address in hourly time periods.

**Table 6–60 MGMT\$CSM\_IP\_HOURLY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR	VARCHAR2(1024)	Node name (if available) or IP address of the requesting target. This may be a gateway or router if one exists between the user machine and the webserver target.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.

**Table 6–60 (Cont.) MGMT\$CSM\_IP\_HOURLY**

Column	Datatype	Description
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze hourly HTTP response time patterns and statistical (min, max, avg, stddev) information for a given visitor IP address of a composite target (For example, what was the avg response time experienced from IP address 192.168.1.1 when accessing the website my.oracle.com between 8 and 9 this morning).

**6.9.14 MGMT\$CSM\_IP\_DAILY**

MGMT\$CSM\_IP\_DAILY displays statistical information about the HTTP requests for a given composite target, aggregated by visitor IP address in daily time periods.

**Table 6–61 MGMT\$CSM\_IP\_DAILY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR	VARCHAR2(1024)	Node name (if available) or IP address of the requesting target. This may be a gateway or router if one exists between the user machine and the webserver target.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze daily HTTP response time patterns and statistical (min, max, avg, stddev) information for a given visitor IP address of a composite target (For example, what was the avg response time experienced from IP address 192.168.1.1 when accessing the website my.oracle.com last Monday).

## 6.9.15 MGMT\$CSM\_IP\_DIST\_HOURLY

MGMT\$CSM\_IP\_DIST\_HOURLY displays the distribution of the HTTP request response times, in seconds, for a visitor IP address of a composite target. The data is aggregated in hourly time periods.

**Table 6–62 MGMT\$CSM\_IP\_DIST\_HOURLY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR	VARCHAR2(1024)	Node name (if available) or IP address of the requesting target. This may be a gateway or router if one exists between the user machine and the webserver target.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	NUMBER	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

### Usage Notes

Analyze the distribution of the HTTP response times for a visitor IP address of a composite target during a specific hour of the day (For example, how many requests from IP address 192.168.1.1 to the website my.oracle.com took longer than 5 seconds between 8 and 9 this morning).

## 6.9.16 MGMT\$CSM\_IP\_DIST\_DAILY

MGMT\$CSM\_IP\_DIST\_DAILY displays the distribution of the HTTP request response times, in seconds, for a visitor IP address of a composite target. The data is aggregated in daily time periods.

**Table 6–63 MGMT\$CSM\_IP\_DIST\_DAILY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR	VARCHAR2(1024)	Node name (if available) or IP address of the requesting target. This may be a gateway or router if one exists between the user machine and the webserver target.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	NUMBER	Name of the metric being defined.

**Table 6–63 (Cont.) MGMT\$CSM\_IP\_DIST\_DAILY**

Column	Datatype	Description
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Usage Notes**

Analyze the distribution of the HTTP response times for a given visitor IP address of a composite target during a specific day (For example, how many requests from IP address 192.168.1.1 to the website my.oracle.com took longer than 5 seconds last Monday).

**6.9.17 MGMT\$CSM\_MT\_IP\_HOURLY**

MGMT\$CSM\_MT\_IP\_HOURLY displays statistical information about the HTTP requests for a given Web Cache target of a composite target, aggregated by visitor IP address in hourly time periods. If a Web Cache target is not a member of any composite targets, the composite target columns are presented as NULL. If a Web Cache target is a member of multiple composite targets, a row for each composite target is presented.

**Table 6–64 MGMT\$CSM\_MT\_IP\_HOURLY**

Column	Datatype	Description
MEMBER_TARGET_NAME	VARCHAR2(64)	The Web Cache target.
MEMBER_TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
COMPOSITE_TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table. May be null if the Web Cache target is not a member of any composite targets.
COMPOSITE_TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target. May be null if the Web Cache target is not a member of any composite targets.
VISITOR	VARCHAR2(1024)	Node name (if available) or IP address of the requesting target. This may be a gateway or router if one exists between the user machine and the webserver target.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.

**Table 6–64 (Cont.) MGMT\$CSM\_MT\_IP\_HOURLY**

Column	Datatype	Description
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze hourly HTTP response time patterns and statistical (min, max, avg, stddev) information for a given visitor IP address of a Web Cache target and optionally of a composite target (For example, what was the avg response time experienced from IP address 192.168.1.1 when accessing Web Cache WEB5 of the website my.oracle.com between 8 and 9 this morning).

**6.9.18 MGMT\$CSM\_MT\_IP\_DAILY**

MGMT\$CSM\_MT\_IP\_DAILY displays statistical information about the HTTP requests for a given Web Cache target of a composite target, aggregated by visitor IP address in daily time periods. If a Web Cache target is not a member of any composite targets, the composite target columns are presented as NULL. If a Web Cache target is a member of multiple composite targets, a row for each composite target is presented.

**Table 6–65 MGMT\$CSM\_MT\_IP\_DAILY**

Column	Datatype	Description
MEMBER_TARGET_NAME	VARCHAR2(64)	The Web Cache target.
MEMBER_TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
COMPOSITE_TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table. May be null if the Web Cache target is not a member of any composite targets.
COMPOSITE_TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target. May be null if the Web Cache target is not a member of any composite targets.
VISITOR	VARCHAR2(1024)	Node name (if available) or IP address of the requesting target. This may be a gateway or router if one exists between the user machine and the webserver target.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.

**Table 6–65 (Cont.) MGMT\$CSM\_MT\_IP\_DAILY**

Column	Datatype	Description
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze daily HTTP response time patterns and statistical (min, max, avg, stddev) information for a given visitor IP address of a Web Cache target and optionally of a composite target (For example, what was the avg response time experienced from IP address 192.168.1.1 when accessing Web Cache WEB5 of the website my.oracle.com last Monday).

**6.9.19 MGMT\$CSM\_MT\_IP\_DIST\_HOURLY**

MGMT\$CSM\_MT\_IP\_DIST\_HOURLY displays the distribution of the HTTP request response times, in seconds, for a visitor IP address of a given Web Cache target of a composite target. The data is aggregated in hourly time periods. If a Web Cache target is not a member of any composite targets, the composite target columns are presented as NULL. If a Web Cache target is a member of multiple composite targets, a row for each composite target is presented.

**Table 6–66 MGMT\$CSM\_MT\_IP\_DIST\_HOURLY**

Column	Datatype	Description
MEMBER_TARGET_NAME	VARCHAR2(64)	The Web Cache target.
MEMBER_TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
COMPOSITE_TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table. May be null if the Web Cache target is not a member of any composite targets.
COMPOSITE_TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target. May be null if the Web Cache target is not a member of any composite targets.
VISITOR	VARCHAR2(1024)	Node name (if available) or IP address of the requesting target. This may be a gateway or router if one exists between the user machine and the webserver target.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Usage Notes**

Analyze the distribution of the HTTP response times for a visitor IP address of a Web Cache target of a composite target during a specific hour of the day (For example, how many requests from IP address 192.168.1.1 to Web Cache WEB5 of the website my.oracle.com took longer than 5 seconds between 8 and 9 this morning).

## 6.9.20 MGMT\$CSM\_MT\_IP\_DIST\_DAILY

MGMT\$CSM\_MT\_IP\_DIST\_DAILY displays the distribution of the HTTP request response times, in seconds, for a visitor IP address of a given Web Cache target of a composite target. The data is aggregated in daily time periods. If a Web Cache target is not a member of any composite targets, the composite target columns are presented as NULL. If a Web Cache target is a member of multiple composite targets, a row for each composite target is presented.

**Table 6–67 MGMT\$CSM\_MT\_IP\_DIST\_DAILY**

Column	Datatype	Description
MEMBER_TARGET_NAME	VARCHAR2(64)	The Web Cache target.
MEMBER_TARGET_TYPE	VARCHAR2(64)	The target type defines the set of metrics that are applicable for the target.
COMPOSITE_TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table. May be null if the Web Cache target is not a member of any composite targets.
COMPOSITE_TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target. May be null if the Web Cache target is not a member of any composite targets.
VISITOR	VARCHAR2(1024)	Node name (if available) or IP address of the requesting target. This may be a gateway or router if one exists between the user machine and the webserver target.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

### Usage Notes

Analyze the distribution of the HTTP response times for a visitor IP address of a Web Cache target of a composite target during a specific day (For example, how many requests from IP address 192.168.1.1 to Web Cache WEB5 of the website my.oracle.com took longer than 5 seconds last Monday).

## 6.9.21 MGMT\$CSM\_DOMAIN\_HOURLY

MGMT\$CSM\_DOMAIN\_HOURLY displays statistical information about the HTTP requests for a given composite target, aggregated by the visitor IP domain in hourly time periods.

**Table 6–68 MGMT\$CSM\_DOMAIN\_HOURLY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.

**Table 6–68 (Cont.) MGMT\$CSM\_DOMAIN\_HOURLY**

Column	Datatype	Description
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR_DOMAIN	VARCHAR2(1024)	Domain of the machine making requests.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze hourly HTTP response time patterns and statistical (min, max, avg, stddev) information for a given visitor IP domain of a composite target (For example, what was the avg response time experienced from users of the domain oracle.co.uk when accessing the website my.oracle.com between 8 and 9 this morning).

**6.9.22 MGMT\$CSM\_DOMAIN\_DAILY**

MGMT\$CSM\_DOMAIN\_DAILY displays statistical information about the HTTP requests for a given composite target, aggregated by visitor IP domain in daily time periods.

**Table 6–69 MGMT\$CSM\_DOMAIN\_DAILY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR_DOMAIN	VARCHAR2(1024)	Domain of the machine making requests.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period



**Table 6–69 (Cont.) MGMT\$CSM\_DOMAIN\_DAILY**

Column	Datatype	Description
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze daily HTTP response time patterns and statistical (min, max, avg, stddev) information for a given visitor IP domain of a composite target (For example, what was the avg response time experienced from users of the domain oracle.co.uk when accessing the website my.oracle.com last Monday).

**6.9.23 MGMT\$CSM\_DOMAIN\_DIST\_HOURLY**

MGMT\$CSM\_DOMAIN\_DIST\_HOURLY displays the distribution of the HTTP request response times, in seconds, for a visitor IP domain of a composite target. The data is aggregated in hourly time periods.

**Table 6–70 MGMT\$CSM\_DOMAIN\_DIST\_HOURLY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR_DOMAIN	VARCHAR2(1024)	Domain of the machine making requests.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Usage Notes**

Analyze the distribution of the HTTP response times for a visitor IP domain of a composite target during a specific hour of the day (For example, how many requests from users of the domain oracle.co.uk to the website my.oracle.com took longer than 5 seconds between 8 and 9 this morning).

## 6.9.24 MGMT\$CSM\_DOMAIN\_DIST\_DAILY

MGMT\$CSM\_DOMAIN\_DIST\_DAILY displays the distribution of the HTTP request response times, in seconds, for a visitor IP domain of a composite target. The data is aggregated in daily time periods.

**Table 6–71 MGMT\$CSM\_DOMAIN\_DIST\_DAILY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR_DOMAIN	VARCHAR2(1024)	Domain of the machine making requests.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

### Usage Notes

Analyze the distribution of the HTTP response times for a given visitor IP domain of a composite target during a specific day (For example, how many requests from users of the domain oracle.co.uk to the website my.oracle.com took longer than 5 seconds last Monday).

## 6.9.25 MGMT\$CSM\_SUBNET\_HOURLY

MGMT\$CSM\_SUBNET\_HOURLY displays statistical information about the HTTP requests for a given composite target, aggregated by the visitor IP subnet in hourly time periods.

**Table 6–72 MGMT\$CSM\_SUBNET\_HOURLY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR_SUBNET	VARCHAR2(32)	Domain subnet of the machine making the requests.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period

**Table 6–72 (Cont.) MGMT\$CSM\_SUBNET\_HOURLY**

Column	Datatype	Description
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze hourly HTTP response time patterns and statistical (min, max, avg, stddev) information for a given visitor IP subnet of a composite target (For example, what was the avg response time experienced from users of the subnet 192.168.1 when accessing the website my.oracle.com between 8 and 9 this morning).

**6.9.26 MGMT\$CSM\_SUBNET\_DAILY**

MGMT\$CSM\_SUBNET\_DAILY displays statistical information about the HTTP requests for a given composite target, aggregated by visitor IP subnet in daily time periods.

**Table 6–73 MGMT\$CSM\_SUBNET\_DAILY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR_SUBNET	VARCHAR2(32)	Domain subnet of the machine making the requests.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.
AVERAGE	NUMBER	The average of the metric values for the samples that have been included in the rollup period
MINIMUM	NUMBER	The minimum value for the metric for the samples that have been included in the rollup period.
MAXIMUM	NUMBER	The maximum value for the metric for samples that have been included in the rollup period.
STANDARD_DEVIATION	NUMBER	The standard deviation for the metric values that have been included in the rollup period.
VARIANCE	NUMBER	The variance for the metric values that has been included in the rollup period

**Usage Notes**

Analyze daily HTTP response time patterns and statistical (min, max, avg, stddev) information for a given visitor IP subnet of a composite target (For example, what was the avg response time experienced from users of the subnet 192.168.1 when accessing the website my.oracle.com last Monday).

**6.9.27 MGMT\$CSM\_SUBNET\_DIST\_HOURLY**

MGMT\$CSM\_SUBNET\_DIST\_HOURLY displays the distribution of the HTTP request response times, in seconds, for a visitor IP subnet of a composite target. The data is aggregated in hourly time periods.

**Table 6–74 MGMT\$CSM\_SUBNET\_DIST\_HOURLY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR_SUBNET	VARCHAR2(32)	Domain subnet of the machine making the requests.
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Usage Notes**

Analyze the distribution of the HTTP response times for a visitor IP subnet of a composite target during a specific hour of the day (For example, how many requests from users of the subnet 192.168.1 to the website my.oracle.com took longer than 5 seconds between 8 and 9 this morning).

**6.9.28 MGMT\$CSM\_SUBNET\_DIST\_DAILY**

MGMT\$CSM\_SUBNET\_DIST\_DAILY displays the distribution of the HTTP request response times, in seconds, for a visitor IP subnet of a composite target. The data is aggregated in daily time periods.

**Table 6–75 MGMT\$CSM\_SUBNET\_DIST\_DAILY**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(64)	The name of the composite target. A composite target is a grouping of one or more target name and target types from the target table.
TARGET_TYPE	VARCHAR2(64)	The type of the composite target. The type defines the set of metrics that are applicable for the target.
VISITOR_SUBNET	VARCHAR2(32)	Domain subnet of the machine making the requests.

**Table 6–75 (Cont.) MGMT\$CSM\_SUBNET\_DIST\_DAILY**

Column	Datatype	Description
ROLLUP_TIMESTAMP	DATE	Start of the rollup period (time zone of the target).
METRIC_NAME	VARCHAR2(64)	Name of the metric being defined.
METRIC_VALUE	NUMBER	Number of milliseconds required to access this url. This value is the difference in milliseconds between SUBMIT_ACTION_TIMESTAMP and LOAD_ACTION_TIMESTAMP
SAMPLE_COUNT	NUMBER	The number of non-NULL samples for the metric that were aggregated.

**Usage Notes**

Analyze the distribution of the HTTP response times for a given visitor IP subnet of a composite target during a specific day (For example, how many requests from users of the subnet 192.168.1 to the website my.oracle.com took longer than 5 seconds last Monday).

**6.9.29 MGMT\$E2E\_1DAY**

MGMT\$E2E\_1DAY displays the E2E data where rollup time spans from 7 days to 31 days.

**Table 6–76 MGMT\$E2E\_1DAY**

Column	Datatype	Description
URI	VARCHAR2(2000)	URL for which the end to end trace is sampled
HITS	NUMBER(8)	No of times the uri is hit for before aggregating the sampled data.
TOTAL_HIT_TIME	NUMBER(10)	Total time taken for the uri to respond.
SERVLET_COUNT	NUMBER(8)	No of servlets used for serving the uri
SERVLET_TIME	NUMBER(10)	Total time taken by all servlets for serving the uri.
JSP_COUNT	NUMBER(8)	No of jsps used for serving the uri.
JSP_TIME	NUMBER(10)	Total time taken by all jsps involved in serving the uri.
EJB_COUNT	NUMBER(8)	No of EJB objects involved in serving the uri.
EJB_TIME	NUMBER(10)	Total time taken by the all the EJB objects involved in the serving the uri.
JDBC_TIME	NUMBER(10)	Total JDBC including SQL, PL/SQL times for serving the uri.
TARGET_GUID	RAW(16)	Target GUID of the web application
ROLLUP_TIMESTAMP	DATE	Since each record in the base table, mgmt_e2e_summary_1day, of this view is a rollup of daily data from mgmt_e2e_summary_1hour, this date corresponds to the time when that rollup has happened.

**6.9.30 MGMT\$E2E\_HOURLY**

MGMT\$E2E\_HOURLY displays E2E data where rollup time spans from 24 hours to last 7 days.

**Table 6–77 MGMT\$E2E\_HOURLY**

Column	Datatype	Description
URI	VARCHAR2(2000)	URL for which the end to end trace is sampled
HITS	NUMBER(8)	No of times the uri is hit for before aggregating the sampled data.
TOTAL_HIT_TIME	NUMBER(10)	Total time taken for the uri to respond.
SERVLET_COUNT	NUMBER(8)	No of servlets used for serving the uri
SERVLET_TIME	NUMBER(10)	Total time taken by all servlets for serving the uri.
JSP_COUNT	NUMBER(8)	No of jsps used for serving the uri.
JSP_TIME	NUMBER(10)	Total time taken by all jsps involved in serving the uri.
EJB_COUNT	NUMBER(8)	No of EJB objects involved in serving the uri.
EJB_TIME	NUMBER(10)	Total time taken by the all the EJB objects involved in the serving the uri.
JDBC_TIME	NUMBER(10)	Total JDBC including SQL, PL/SQL times for serving the uri.
TARGET_GUID	RAW(16)	Target GUID of the web application
ROLLUP_TIMESTAMP	DATE	Since each record in the base table, mgmt_e2e_summary_1day, of this view is a rollup of daily data from mgmt_e2e_summary_1hour, this date corresponds to the time when that rollup has happened.

### 6.9.31 MGMT\$E2E\_RAW

MGMT\$E2E\_RAW displays E2E data where rollup time is less than 24 hours.

**Table 6–78 MGMT\$E2E\_RAW**

Column	Datatype	Description
URI	VARCHAR2(2000)	URL for which the end to end trace is sampled
HITS	NUMBER(8)	No of times the uri is hit for before aggregating the sampled data.
TOTAL_HIT_TIME	NUMBER(10)	Total time taken for the uri to respond.
SERVLET_COUNT	NUMBER(8)	No of servlets used for serving the uri
SERVLET_TIME	NUMBER(10)	Total time taken by all servlets for serving the uri.
JSP_COUNT	NUMBER(8)	No of jsps used for serving the uri.
JSP_TIME	NUMBER(10)	Total time taken by all jsps involved in serving the uri.
EJB_COUNT	NUMBER(8)	No of EJB objects involved in serving the uri.
EJB_TIME	NUMBER(10)	Total time taken by the all the EJB objects involved in the serving the uri.
JDBC_TIME	NUMBER(10)	Total JDBC including SQL, PL/SQL times for serving the uri.
TARGET_GUID	RAW(16)	Target GUID of the web application
ROLLUP_TIMESTAMP	DATE	Since each record in the base table, mgmt_e2e_summary_1day, of this view is a rollup of daily data from mgmt_e2e_summary_1hour, this date corresponds to the time when that rollup has happened.

## 6.10 Configuration Views

### 6.10.1 MGMT\$DB\_TABLESPACES

MGMT\$DB\_TABLESPACES displays configuration settings for tablespaces. Tablespace settings are collected from the sys.dba\_tablespaces, dba\_free\_space, dba\_data\_files, dba\_temp\_files, and v\$temp\_extent\_pool tables..

**Table 6–79 MGMT\$DB\_TABLESPACES**

Column	Datatype	Description
HOST	VARCHAR2(256)	Name of the target where the metrics will be collected.
TARGET_NAME	VARCHAR2(256)	Name of the database containing the datafiles.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace.
CONTENTS	VARCHAR2(9)	Tablespace contents: PERMANENT or TEMPORARY.
STATUS	VARCHAR2(10),	Tablespace status: ONLINE, OFFLINE, or READ ONLY.
EXTENT_MANAGEMENT	VARCHAR2(10),	Extent management tracking: DICTIONARY or LOCAL.
ALLOCATION_TYPE	VARCHAR2(10),	Type of extent allocation in effect for this tablespace.
LOGGING	VARCHAR2(10),	Default logging attribute.
TABLESPACE_SIZE	NUMBER	Current size of the tablespace in bytes.
INITIAL_EXT_SIZE	NUMBER	Default initial extent size
NEXT_EXTENT	NUMBER	Next extent in the sequence.
INCREMENT_BY	NUMBER	Default percent increase for extent size.
MAX_EXTENTS	NUMBER	Default maximum number of extents
TABLESPACE_USED_SIZE	NUMBER	Amount of data (in bytes) contained in the tablespace.
SEGMENT_SPACE_MANAGEMENT	VARCHAR2(6)	Indicates whether the free and used segment space in the tablespace is managed using free lists (MANUAL) or bitmaps (AUTO)
BLOCK_SIZE	NUMBER	Tablespace block size
MIN_EXTENTS	NUMBER	Default minimum number of extents
MIN_EXTLEN	NUMBER	Minimum extent size for this tablespace
BIGFILE	VARCHAR2(3)	Indicates whether the tablespace is a bigfile tablespace (YES) or a smallfile tablespace (NO)

#### Usage Notes

Obtain control file configuration settings across all managed database targets.

## 6.10.2 MGMT\$DB\_DATAFILES

MGMT\$DB\_DATAFILES displays the configuration settings for datafiles. The datafile settings are collected from sources such as sys.dba\_data\_files, v\$datafile, sys.dba\_free\_space, sys.dba\_tablespaces, sys.dba\_temp\_files, v\$tempfile.

**Table 6–80 MGMT\$DB\_DATAFILES**

Column	Datatype	Description
HOST	VARCHAR2(256)	Name of the target where the metrics will be collected.
TARGET_NAME	VARCHAR2(256)	Name of the database containing the datafiles.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
FILE_NAME	VARCHAR2(512)	Name of the datafile.
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace containing
STATUS	VARCHAR2(10)	Datafile status: ACTIVE or NOT ACTIVE
FILE_SIZE	NUMBER	Size of the datafile
AUTOEXTENSIBLE	VARCHAR2(3)	Autoextensible indicator
INCREMENT_BY	NUMBER	Autoextension increment.
MAX_FILE_SIZE	NUMBER	Maximum file size in bytes
OS_STORAGE_ENTITY	VARCHAR2(512)	OS level storage entity on which the file resides. For regular files it is the name of the filesystem on which the file resides. For character or raw files it is the name of the raw device

## 6.10.3 MGMT\$DB\_CONTROLFILES

MGMT\$DB\_CONTROLFILES displays the configuration settings for database control files.

**Table 6–81 MGMT\$DB\_CONTROLFILES**

Column	Datatype	Description
HOST	VARCHAR2(256)	Name of the target where the metrics will be collected.
TARGET_NAME	VARCHAR2(256)	Name of the database containing the datafiles.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
FILE_NAME	VARCHAR2(512)	Name of the database control file.



**Table 6–81 (Cont.) MGMT\$DB\_CONTROLFILES**

Column	Datatype	Description
STATUS	VARCHAR2(10),	The type of control file: STANDBY - indicates database is in standby mode LOGICAL - indicates the database is a logical standby database (not a physical standby) CLONE - indicates a clone database BACKUP   CREATED - indicates database is being recovered using a backup or created control file. CURRENT - the control file changes to this type following a standby database activate or database open after recovery.
CREATION_DATE	DATE	Control file creation date.
SEQUENCE_NUM	NUMBER	Control file sequence number incremented by control file transactions.
CHANGE_NUM	NUMBER	Last change number in the backup control file. Value is NULL if the control file is not a backup.
MOD_DATE	DATE	Last timestamp in the backup control file. NULL if the control file is not a backup.
OS_STORAGE_ENTITY	VARCHAR2(512)	OS level storage entity on which the file resides. For regular files it is the name of the filesystem on which the file resides. For character or raw files it is the name of the raw device

#### 6.10.4 MGMT\$DB\_DBNINSTANCEINFO

MGMT\$DB\_DBNINSTANCEINFO displays general information about database instance. The instance information is collected from v\$database, v\$version, v\$instance, global\_name, database\_properties and v\$nls\_parameters.

**Table 6–82 MGMT\$DB\_DBNINSTANCEINFO**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	Name of the target host where the metrics will be collected.
TARGET_NAME	VARCHAR2(256)	Name of the database target from which the metrics are collected.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
DATABASE_NAME	VARCHAR2(9)	Name of the database.
GLOBAL_NAME	VARCHAR2(4000)	Global name of the database
BANNER	VARCHAR2(64)	Component name and version number
HOST	VARCHAR2(64)	Name of the host machine.
INSTANCE_NAME	VARCHAR2(16)	Name of the instance.
STARTUP_TIME	DATE	Time when instance was started up.
LOGINS	VARCHAR2(10)	ALLOWED or RESTRICTED
LOG_MODE	VARCHAR2(12)	The archive log mode, either ARCHIVELOG or NOARCHIVELOG.

**Table 6–82 (Cont.) MGMT\$DB\_DBNINSTANCEINFO**

Column	Datatype	Description
OPEN_MODE	VARCHAR2(10)	Open mode information.
DEFAULT_TEMP_TABLESPACE	VARCHAR2(30)	Default temporary tablespace name.
CHARACTERSET	VARCHAR2(64)	NLS parameter value for NLS_CHARACTERSET
NATIONAL_CHARACTERSET	VARCHAR2(64)	NLS parameter value for NLS_NCHAR_CHARACTERSET

**Usage Notes**

Obtain general instance information across all database targets.

**6.10.5 MGMT\$DB\_FEATUREUSAGE**

MGMT\$DB\_FEATUREUSAGE displays information about database feature usage.

**Table 6–83 MGMT\$DB\_FEATUREUSAGE**

Column	Datatype	Description
HOST	VARCHAR2(256)	Name of the host target where the database feature usage information is collected.
DATABASE_NAME	VARCHAR2(256)	Name of the database where the database feature usage information is collected.
INSTANCE_NAME	VARCHAR2(16)	Name of the instance where the database feature usage information is collected.
TARGET_TYPE	VARCHAR2(64)	Either Oracle_database or rac_database.
DBID	NUMBER	A unique number that identifies a database instance.
NAME	VARCHAR2(64)	The feature name.
CURRENTLY_USED	VARCHAR2(5)	TRUE if the feature is currently in use, FALSE if the feature is not in use.
DETECTED_USAGES	NUMBER	The number of times the feature has been used by the database.
FIRST_USAGE_DATE	DATE	The date that the first usage of the feature occurred.
LAST_USAGE_DATE	DATE	The date of the most recent usage of the feature.
VERSION	VARCHAR2(17)	The version number of the database.
LAST_SAMPLE_DATE	DATE	The date that the database was last evaluated for feature usage.
LAST_SAMPLE_PERIOD	NUMBER	The interval between the LAST_SAMPLE_DATE date and the database feature usage evaluation before that (by default, seven days).
SAMPLE_INTERVAL	NUMBER	The number of seconds between the LAST_SAMPLE_DATE date and the next database feature usage evaluation.
TOTAL_SAMPLES	NUMBER	The total number of database feature usage evaluation samples that have been collected.
AUX_COUNT	NUMBER	For Oracle internal use only.
DESCRIPTION	VARCHAR2(128)	The description of the feature.

**Usage Notes**

This view can be used to gain an enterprise-wide view of database feature usage across all Oracle databases.

**6.10.6 MGMT\$DB\_INIT\_PARAMS**

MGMT\$DB\_INIT\_PARAMS displays initialization parameter settings for the database. Initialization parameter settings are collected from v\$parameter.

**Table 6–84 MGMT\$DB\_INIT\_PARAMS**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	Name of the target where the metrics will be collected.
TARGET_NAME	VARCHAR2(256)	Name of the database target from which the metrics are collected.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
NAME	VARCHAR2(64)	Name of the initialization parameter.
ISDEFAULT	VARCHAR2(6)	Indicates whether the parameter value is the default.
VALUE	VARCHAR2(512)	The parameter value.
DATATYPE	VARCHAR2	The data type that the value string can be mapped to, for example, NUMBER, DATE, or TEXT.'

**Usage Notes**

Obtain initialization parameter settings across all database targets.

**6.10.7 MGMT\$DB\_LICENSE**

MGMT\$DB\_LICENSE displays database license configuration settings. Database license configuration settings are collected from v\$license.

**Table 6–85 MGMT\$DB\_LICENSE**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	The name of the host on which the database is running.
TARGET_NAME	VARCHAR2(256)	Name of the database containing the tablespace.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
SESSIONS_MAX	NUMBER	The maximum number of sessions allowed for the database.
SESSIONS_WARNING	NUMBER	The number of sessions which will generate a warning for the database.
SESSIONS_CURRENT	NUMBER	The current number of sessions for the database.

**Table 6–85 (Cont.) MGMT\$DB\_LICENSE**

Column	Datatype	Description
SESSIONS_HIGHWATER	NUMBER	The highest water mark of sessions for the database.
USERS_MAX	NUMBER	The maximum number of users for the database.

**Usage Notes**

This view can be used to obtain database license configuration settings across all database targets.

**6.10.8 MGMT\$DB\_REDOLOGS**

MGMT\$DB\_REDOLOGS displays redo log configuration settings for the database. Redo log configuration settings are collected from the v\$log and v\$logfile tables.

**Table 6–86 MGMT\$DB\_REDOLOGS**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	Name of the target where the metrics will be collected.
TARGET_NAME	VARCHAR2(256)	Name of the database target from which the metrics are collected.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
GROUP_NUM	NUMBER	Redo log group identifier number.
STATUS	VARCHAR2(16),	og status: UNUSED - The online redo log has never been written to. This is the state of a redo log that was just added, or just after a RESETLOGS, when it is not the current redo log. CURRENT - This is the current redo log. This implies that the redo log is active. The redo log could be open or closed. ACTIVE - The log is active but is not the current log. It is needed for crash recovery. It may be in use for block recovery. It might or might not be archived. CLEARING - The log is being re-created as an empty log after an ALTER DATABASE CLEAR LOGFILE statement. After the log is cleared, the status changes to UNUSED. CLEARING_CURRENT - The current log is being cleared of a closed thread. The log can stay in this status if there is some failure in the switch such as an I/O error writing the new log header. INACTIVE - The log is no longer needed for instance recovery. It may be in use for media recovery. It might or might not be archived.
MEMBERS	NUMBER	Number of members in the log group
FILE_NAME	VARCHAR2(513),	Redo log file (member) name.
ARCHIVED	VARCHAR2(3),	Archive status either YES or NO.

**Table 6–86 (Cont.) MGMT\$DB\_REDOLOGS**

Column	Datatype	Description
LOGSIZE	NUMBER	Size of the log file in bytes
SEQUENCE_NUM	NUMBER	Log sequence number
FIRST_CHANGE_SCN	NUMBER	Lowest SCN in the log.
OS_STORAGE_ENTITY	VARCHAR2(512)	OS level storage entity on which the file resides. For regular files it is the name of the filesystem on which the file resides. For character or raw files it is the name of the raw device.
THREAD_NUM	NUMBER	Log thread number.

**Usage Notes**

Obtain redo log group / file configuration settings across all database targets.

**6.10.9 MGMT\$DB\_ROLLBACK\_SEGS**

MGMT\$DB\_ROLLBACK\_SEGS displays rollback segments configuration settings for the database. Rollback segments configuration settings are collected from the sys.dba\_rollback\_segs and v\$rollstat tables.

**Table 6–87 MGMT\$DB\_ROLLBACK\_SEGS**

Column	Datatype	Description
HOST_NAME	VARCHAR2	Name of the target where the metrics will be collected.
TARGET_NAME	VARCHAR2(256)	Name of the database containing the datafiles.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
ROLLNAME	VARCHAR2(64)	Name of the rollback segment
STATUS	VARCHAR2(10)	Rollback segment status
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace containing the rollback segment.
EXTENTS	NUMBER	Number of extents in rollback segment.
ROLLSIZE	NUMBER	Size in bytes of rollback segment. This values differs by the number of bytes in one database block from the value of the BYTES column of the ALL/DBA/USER_SEGMENTS views.
INITIAL_SIZE	NUMBER	Initial extent size in bytes.
NEXT_SIZE	NUMBER	Secondary extent size in bytes.
MAXIMUM_EXTENTS	NUMBER	Maximum number of extents.
MINIMUM_EXTENTS	NUMBER	Minimum number of extents.
PCT_INCREASE	NUMBER	Percent increase for extent size.
OPTSIZE	NUMBER	Optimal size for rollback segments.
AVEACTIVE	NUMBER	Current size of active extents averaged over time.
WRAPS	NUMBER	Number of times rollback segment is wrapped.

**Table 6–87 (Cont.) MGMT\$DB\_ROLLBACK\_SEGS**

Column	Datatype	Description
SHRINKS	NUMBER	Number of times the size of a rollback segment decreases.
AVESHINK	NUMBER	Average shrink size.
HWMSIZE	NUMBER	High water mark of rollback segment size.

**Usage Notes**

Obtain rollback segments configuration settings across all database targets.

**6.10.10 MGMT\$DB\_SGA**

MGMT\$DB\_SGA displays System Global Area (SGA) configuration settings. SGA settings are collected from the v\$sga and v\$sgastat tables.

**Table 6–88 MGMT\$DB\_SGA**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	Name of the target where the metrics will be collected.
TARGET_NAME	VARCHAR2(256)	Name of the database containing the datafiles.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_ database.
TARGET_GUID	RAW(16)	The unique ID for the database target
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
SGANAME	VARCHAR2(64)	SGA component name.
SGASIZE	NUMBER	SGA component size in kilobytes or megabytes

**Usage Notes**

Obtain System Global Area configuration settings across all database targets.

**6.10.11 MGMT\$DB\_TABLESPACES**

MGMT\$DB\_TABLESPACES displays configuration settings for tablespaces. Tablespace settings are collected from the sys.dba\_tablespaces, dba\_free\_space, dba\_data\_files, dba\_temp\_files, and v\$temp\_extent\_pool tables.

**Table 6–89 MGMT\$DB\_TABLESPACES**

Column	Datatype	Description
HOST	VARCHAR2(256)	Name of the target where the metrics will be collected.
TARGET_NAME	VARCHAR2(64)	The type of target, for example, Oracle_ database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
TABLESPACE_NAME	VARCHAR2(30)	Name of the tablespace.
CONTENTS	VARCHAR2(9)	Tablespace contents: PERMANENT or TEMPORARY.

**Table 6–89 (Cont.) MGMT\$DB\_TABLESPACES**

Column	Datatype	Description
STATUS	VARCHAR2(10),	Tablespace status: ONLINE, OFFLINE, or READ ONLY.
EXTENT_MANAGEMENT	VARCHAR2(10),	Extent management tracking: DICTIONARY or LOCAL.
ALLOCATION_TYPE	VARCHAR2(10),	Type of extent allocation in effect for this tablespace.
LOGGING	VARCHAR2(10),	Default logging attribute.
TABLESPACE_SIZE	NUMBER	Current size of the tablespace in bytes.
INITIAL_EXT_SIZE	NUMBER	Default initial extent size
INCREMENT_BY	NUMBER	Default percent increase for extent size.
MAX_EXTENTS	NUMBER	Default maximum number of extents

**Usage Notes**

Obtain tablespace configuration settings across all database targets.

**6.10.12 MGMT\$DB\_OPTIONS**

MGMT\$DB\_OPTIONS displays whether or not the option is currently LOADED and ACTIVE, or either the option does not exist or is NOT LOADED or INACTIVE. Options settings are collected by checking user name and status in the sys.dba\_users and dba\_registry tables.

**Table 6–90 MGMT\$DB\_OPTIONS**

Column	Datatype	Description
HOST	VARCHAR2(256)	Name of the target where the metrics will be collected.
TARGET_NAME	VARCHAR2(256)	Name of the database containing the datafiles.
TARGET_TYPE	VARCHAR2(64)	The type of target, for example, Oracle_database.
TARGET_GUID	RAW(16)	The unique ID for the database target.
COLLECTION_TIMESTAMP	DATE	The date and time when the metrics were collected.
NAME	VARCHAR2(30)	Name of the database option.
SELECTED	VARCHAR2(5)	If the option is currently LOADED and ACTIVE (TRUE), or either the option does not exist or is NOT LOADED or INACTIVE (FALSE)

**Usage Notes**

Obtain tablespace configuration settings across all database targets.

**6.11 Oracle Home Patching Views**

### 6.11.1 MGMT\$EM\_HOMES\_PLATFORM

MGMT\$EM\_HOMES\_PLATFORM displays the platform information about the Homes. If the home does not have an ARU platform id, then the platform of the Operating System is considered as the Platform of the home.

**Table 6–91 MGMT\$EM\_HOMES\_PLATFORM**

Column	Datatype	Description
HOME_ID	RAW(16)	Unique id for home
PLATFORM_ID	NUMBER(10)	If the home has an ARU platform it is used, else the platform id of the host is picked
PLATFORM	VARCHAR2(1024)	The platform corresponding to the platform_id

### 6.11.2 MGMT\$HOMES\_AFFECTED

MGMT\$HOMES\_AFFECTED displays the list of homes, vulnerable to bugs, which are fixed by the critical patches released. The number of alerts which are applicable to the home are calculated.

**Table 6–92 MGMT\$HOMES\_AFFECTED**

Column	Datatype	Description
HOST	VARCHAR2(1024))	Host name
HOME_DIRECTORY	VARCHAR2(1024)	Home directory location
TARGET_GUID	RAW(16)	Unique id for target
ALERTS	NUMBER	Number of alerts for this home

### 6.11.3 MGMT\$APPL\_PATCH\_AND\_PATCHSET

MGMT\$APPL\_PATCH\_AND\_PATCHSET displays the list of interim patches and patchsets that are applicable to the Homes.

**Table 6–93 MGMT\$APPL\_PATCH\_AND\_PATCHSET**

Column	Datatype	Description
PATCH_ID	NUMBER	The patch Id
TYPE	VARCHAR2(32)	Patch/Patchset
PRODUCT	VARCHAR2(50)	The product on pertaining to the patch.
PATCH_RELEASE	VARCHAR2(30)	Release version.
PLATFORM	VARCHAR2(40)	Platform on which patch is applicable
ADVISORY	VARCHAR2(256)	The alert name.
HOST_NAME	VARCHAR2(256)	Host name
HOME_LOCATION	VARCHAR2(128)	Home directory location.
PATCH_GUID	RAW(16)	Unique id for the patch/patchset.
TARGET_GUID	RAW(16)	Unique id for target.

### 6.11.4 MGMT\$APPLIED\_PATCHES

MGMT\$APPLIED\_PATCHES displays the list of patches that have been applied on the homes along with the installation time. Each patch can fix more than one bug. The bugs are listed in a comma-separated string.



**Table 6–94** *MGMT\$APPLIED\_PATCHES*

Column	Datatype	Description
PATCH	VARCHAR2(128)	Patch name.
BUGS	VARCHAR2(256)	The bugs fixed by this patch.
INSTALLATION_TIME	DATE	Time of Installation. Timezone of the target.
HOST	VARCHAR2(256)	Host name.
HOME_LOCATION	VARCHAR2(128)	Home location.
HOME_NAME	VARCHAR2(64)	Name of the home.
CONTAINER_GUID	RAW(16)	Name of the home.
TARGET_GUID	RAW(16)	Unique id for target.

### 6.11.5 MGMT\$APPLIED\_PATCHSETS

MGMT\$APPLIED\_PATCHSETS displays the list of patchsets that have been applied on the Homes along with the installation time.

**Table 6–95** *MGMT\$APPLIED\_PATCHSETS*

Column	Datatype	Description
VERSION	VARCHAR2(64)	The version to which the home will get upgraded to when this patchset is applied
NAME	VARCHAR2(128)	Patchset external name
TIMESTAMP	DATE	Time of Installation. Timezone of the target
HOST	VARCHAR2(256)	Host name.
HOME_LOCATION	VARCHAR2(128)	Home location.
HOME_NAME	VARCHAR2(64)	Name of the home.
CONTAINER_GUID	RAW(16)	Name of the home.
TARGET_GUID	RAW(16)	Unique id for target.

## 6.12 Linux Patching Views

### 6.12.1 MGMT\$HOSTPATCH\_HOSTS

MGMT\$HOSTPATCH\_HOSTS displays information required to generate compliance reports.

**Table 6–96** *MGMT\$HOSTPATCH\_HOSTS*

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	Host name
GROUP_NAME	VARCHAR2(256)	The group the host belongs to.
OUT_OF_DATE_PACKAGES	NUMBER(6)	Number of Packages which have a newer version available
ROGUE_PACKAGES	NUMBER(6)	The packages that are not supposed to be installed on the host

## 6.12.2 MGMT\$HOSTPATCH\_GROUPS

MGMT\$HOSTPATCH\_GROUPS displays additional information about a group, the maturity level which is set by the administrator and the packages which need the host to be rebooted on application.

**Table 6–97 MGMT\$HOSTPATCH\_GROUPS**

Column	Datatype	Description
GROUP_NAME	VARCHAR2(256)	The (unique) name of the Group
MATURITY_LEVEL	VARCHAR2(32)	The maturity level of the group. This is set by the administrator.
NEED_REBOOT_PKGS	VARCHAR2(256)	Comma seperated list of packages which need the machine to be rebooted on application

## 6.12.3 MGMT\$HOSTPATCH\_GRP\_COMPL\_HIST

MGMT\$HOSTPATCH\_GRP\_COMPL\_HIST displays information required to generate compliance history reports.

**Table 6–98 MGMT\$HOSTPATCH\_GRP\_COMPL\_HIST**

Column	Datatype	Description
GROUP_NAME	VARCHAR2(256)	Name of the Group
TOTAL_HOSTS	NUMBER(6)	Number of hosts in the Group
COMPLIANT_HOSTS	NUMBER(6)	Number of compliant Hosts in the Group.
LAST_CHECKED_ON	DATE	Date on which this record was collected.

## 6.12.4 MGMT\$HOSTPATCH\_HOST\_COMPL

MGMT\$HOSTPATCH\_HOST\_COMPL displays information required to generate advisory reports.

**Table 6–99 MGMT\$HOSTPATCH\_HOST\_COMPL**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	Host name
PKG_NAME	VARCHAR2(256)	Package name
VERSION	VARCHAR2(64)	Version of the package
IS_OUT_OF_DATE	NUMBER(1)	If out of date
IS_ROGUE	NUMBER(1)	If it is rogue

## 6.13 Security Views

### 6.13.1 MGMT\$ESA\_ALL\_PRIVS\_REPORT

MGMT\$ESA\_ALL\_PRIVS\_REPORT displays a table containing users and roles that have the 'GRANT ANY' privilege in database security reports.

**Table 6–100 MGMT\$ESA\_ALL\_PRIVS\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted a privilege which amounts to all privileges on the database.
OBJECT_NAME	VARCHAR2(512)	The privilege (GRANT ANY PRIVILEGE) if granted directly, or the role through it has been granted.

### 6.13.2 MGMT\$ESA\_ANY\_DICT\_REPORT

MGMT\$ESA\_ANY\_DICT\_REPORT displays a table and a chart containing users and roles with access to any dictionary in database security reports.

**Table 6–101 MGMT\$ESA\_ANY\_DICT\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted a privilege which provides it access to any dictionary in the database.
OBJECT_NAME	VARCHAR2(512)	The privilege if granted directly, or the role through it has been granted.

### 6.13.3 MGMT\$ESA\_ANY\_PRIV\_REPORT

MGMT\$ESA\_ANY\_PRIV\_REPORT displays a table and a chart containing users with 'ANY' in some privilege granted to them in database security reports.

**Table 6–102 MGMT\$ESA\_ANY\_PRIV\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted a privilege having 'ANY'
OBJECT_NAME	VARCHAR2(512)	The privilege if granted directly, or the role through it has been granted.

### 6.13.4 MGMT\$ESA\_AUDIT\_SYSTEM\_REPORT

MGMT\$ESA\_AUDIT\_SYSTEM\_REPORT displays a table containing users and roles with the 'AUDIT SYSTEM' privilege in database security reports.

**Table 6–103 MGMT\$ESA\_AUDIT\_SYSTEM\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted 'AUDIT SYSTEM' privilege.
OBJECT_NAME	VARCHAR2(512)	The privilege if granted directly, or the role through it has been granted.

### 6.13.5 MGMT\$ESA\_BECOME\_USER\_REPORT

MGMT\$ESA\_BECOME\_USER\_REPORT displays a table containing users and roles with the 'BECOME USER' privilege in database security reports.

**Table 6–104 MGMT\$ESA\_BECOME\_USER\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted 'BECOME USER' privilege.
OBJECT_NAME	VARCHAR2(512)	The privilege if granted directly, or the role through it has been granted.

### 6.13.6 MGMT\$ESA\_CATALOG\_REPORT

MGMT\$ESA\_CATALOG\_REPORT displays a table and a chart containing all the users that have a role such as '%CATALOG%' in database security reports.

**Table 6–105 MGMT\$ESA\_CATALOG\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted a role like '%CATALOG%'
OBJECT_NAME	VARCHAR2(512)	The role if granted directly, or the role through it has been granted.

### 6.13.7 MGMT\$ESA\_CONN\_PRIV\_REPORT

MGMT\$ESA\_CONN\_PRIV\_REPORT displays a table and a chart containing users and roles with the CONNECT or RESOURCE role in database security reports.

**Table 6–106 MGMT\$ESA\_CONN\_PRIV\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.

**Table 6–106 (Cont.) MGMT\$ESA\_CONN\_PRIV\_REPORT**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted the CONNECT or RESOURCE role.
OBJECT_NAME	VARCHAR2(512)	The role if granted directly, or the role through it has been granted.

### 6.13.8 MGMT\$ESA\_CREATE\_PRIV\_REPORT

MGMT\$ESA\_CREATE\_PRIV\_REPORT displays a table and a chart containing users and roles with the CREATE privilege in database security reports.

**Table 6–107 MGMT\$ESA\_CREATE\_PRIV\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted the privilege to create an object in the database.
OBJECT_NAME	VARCHAR2(512)	The role if granted directly, or the role through it has been granted.

### 6.13.9 MGMT\$ESA\_DBA\_GROUP\_REPORT

MGMT\$ESA\_DBA\_GROUP\_REPORT displays a table containing members of the operating system user group DBA in database security reports.

**Table 6–108 MGMT\$ESA\_DBA\_GROUP\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The operating system user which is in the user group DBA.
OBJECT_NAME	VARCHAR2(512)	DBA Group

### 6.13.10 MGMT\$ESA\_DBA\_ROLE\_REPORT

MGMT\$ESA\_DBA\_ROLE\_REPORT displays a table containing users and roles with the DBA role granted to them in database security reports.

**Table 6–109 MGMT\$ESA\_DBA\_ROLE\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.

**Table 6–109 (Cont.) MGMT\$ESA\_DBA\_ROLE\_REPORT**

Column	Datatype	Description
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted the DBA Role.
OBJECT_NAME	VARCHAR2(512)	The role if granted directly, or the role through it has been granted.

### 6.13.11 MGMT\$ESA\_DIRECT\_PRIV\_REPORT

MGMT\$ESA\_DIRECT\_PRIV\_REPORT displays a table and a chart containing privileges granted directly in database security reports.

**Table 6–110 MGMT\$ESA\_DIRECT\_PRIV\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	User which has been granted a privilege directly i.e. not via a role.
OBJECT_NAME	VARCHAR2(512)	The privilege that has been granted directly..

### 6.13.12 MGMT\$ESA\_EXMPT\_ACCESS\_REPORT

MGMT\$ESA\_EXMPT\_ACCESS\_REPORT displays a table containing users and roles with the EXEMPT ACCESS POLICY privilege in database security reports.

**Table 6–111 MGMT\$ESA\_EXMPT\_ACCESS\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted the 'EXEMPT ACCESS POLICY' privilege.
OBJECT_NAME	VARCHAR2(512)	The privilege if granted directly, or the role through it has been granted.

### 6.13.13 MGMT\$ESA\_KEY\_OBJECTS\_REPORT

MGMT\$ESA\_KEY\_OBJECTS\_REPORT displays a table and a chart containing users and roles with access to key objects in database security reports.

**Table 6–112 MGMT\$ESA\_KEY\_OBJECTS\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
USER	VARCHAR2(512)	The user which has access to key objects.
OBJECT_NAME	VARCHAR2(23)	The key object to which that use has access.

**Table 6–112 (Cont.) MGMT\$ESA\_KEY\_OBJECTS\_REPORT**

Column	Datatype	Description
PRIVILEGE	VARCHAR2(512)	The privilege on the key object that has been granted to the user.

### 6.13.14 MGMT\$ESA\_OH\_OWNERSHIP\_REPORT

MGMT\$ESA\_OH\_OWNERSHIP\_REPORT displays a table containing file ownership by Oracle Home in database security reports.

**Table 6–113 MGMT\$ESA\_OH\_OWNERSHIP\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The file whose owner is not the ORACLE HOME owner.
OBJECT_NAME	VARCHAR2(512)	The owner of the file.

### 6.13.15 MGMT\$ESA\_OH\_PERMISSION\_REPORT

MGMT\$ESA\_OH\_PERMISSION\_REPORT displays a table containing file permissions by Oracle Home in database security reports.

**Table 6–114 MGMT\$ESA\_OH\_PERMISSION\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The file that has an insecure permission.
OBJECT_NAME	VARCHAR2(512)	The permission of the file.

### 6.13.16 MGMT\$ESA\_POWER\_PRIV\_REPORT

MGMT\$ESA\_POWER\_PRIV\_REPORT displays a table and a chart containing all the users and roles with ALTER SESSION, ALTER SYSTEM, CREATE PROCEDURE or CREATE LIBRARY privileges in database security reports.

**Table 6–115 MGMT\$ESA\_POWER\_PRIV\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user that has powerful privileges.
OBJECT_NAME	VARCHAR2(512)	The powerful privilege held by the user.

### 6.13.17 MGMT\$ESA\_PUB\_PRIV\_REPORT

MGMT\$ESA\_PUB\_PRIV\_REPORT displays a table and a chart containing privileges granted to PUBLIC in database security reports.

**Table 6–116 MGMT\$ESA\_PUB\_PRIV\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The object on which some privilege has been granted to PUBLIC.
OBJECT_NAME	VARCHAR2(512)	The privilege on the object which has been granted to PUBLIC.

### 6.13.18 MGMT\$ESA\_SYS\_PUB\_PKG\_REPORT

MGMT\$ESA\_SYS\_PUB\_PKG\_REPORT displays a table containing system packages with public execute privileges in database security reports.

**Table 6–117 MGMT\$ESA\_SYS\_PUB\_PKG\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	PUBLIC
OBJECT_NAME	VARCHAR2(512)	The package owned by SYS on which PUBLIC has execute privileges.

### 6.13.19 MGMT\$ESA\_TABSP\_OWNERS\_REPORT

MGMT\$ESA\_TABSP\_OWNERS\_REPORT displays a table containing tablespaces and their owners in database security reports.

**Table 6–118 MGMT\$ESA\_TABSP\_OWNERS\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The tablespace.
OBJECT_NAME	VARCHAR2(512)	The owner of the tablespace.

### 6.13.20 MGMT\$ESA\_TRC\_AUD\_PERM\_REPORT

MGMT\$ESA\_TRC\_AUD\_PERM\_REPORT displays a table containing trace and audit files permissions in database security reports.



**Table 6–119 MGMT\$ESA\_TRC\_AUD\_PERM\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The file path.
OBJECT_NAME	VARCHAR2(512)	The purpose of the file.
PERMISSION	VARCHAR2(512)	Permission of the file.

### 6.13.21 MGMT\$ESA\_WITH\_ADMIN\_REPORT

MGMT\$ESA\_WITH\_ADMIN\_REPORT displays a table and a chart containing users and roles having some privileges granted to them with the WITH ADMIN option in database security reports.

**Table 6–120 MGMT\$ESA\_WITH\_ADMIN\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted a privilege 'WITH ADMIN' option.
OBJECT_NAME	VARCHAR2(512)	The privilege which has been granted 'WITH ADMIN' option.

### 6.13.22 MGMT\$ESA\_WITH\_GRANT\_REPORT

MGMT\$ESA\_WITH\_GRANT\_REPORT displays a table and a chart containing users and roles having some privileges granted to them with 'WITH GRANT' option in database security reports.

**Table 6–121 MGMT\$ESA\_WITH\_GRANT\_REPORT**

Column	Datatype	Description
TARGET_GUID	RAW(16)	The GUID of the target for which the report has the data.
TARGET_NAME	VARCHAR2(256)	The name of the target for which the report has the data.
PRINCIPAL	VARCHAR2(512)	The user or role which has been granted a privilege 'WITH GRANT' option.
OBJECT_NAME	VARCHAR2(512)	The privilege which has been granted 'WITH GRANT' option.

## 6.14 Configuration Management Views

### 6.14.1 MGMT\$CSA\_COLLECTIONS

MGMT\$CSA\_COLLECTIONS displays top-level information about all client configurations.

**Table 6–122 MGMT\$CSA\_COLLECTIONS**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
CSACLIENT	VARCHAR2(4000)	The display name plus the custom keys, if they exist
COLLECTION_TIMESTAMP	DATE	The time at which the data was collected from the client
NET_IP	VARCHAR2(20)	The actual IP address of the client.
NET_EFFECTIVE_IP	VARCHAR2(20)	The client IP address seen by the server
COLLECTION_MESSAGE	VARCHAR2(4000)	Error message generated while applet was running
OS_USER_NAME	VARCHAR2(500)	The client's OS username.
HOSTNAME	VARCHAR2(128)	The client's hostname.
DOMAIN	VARCHAR2(500)	The client's domain.
BOOT_DISK_VOLUME_SERIAL_NUM	VARCHAR2(629)	The client's boot disk volume serial number
COMPLIANCE	NUMBER	The overall compliance score for the client (15=passed, 18=info, 20=warning, 25=critical)
APPID	VARCHAR2(128)	The collection tag for this client configuration
NET_SUBNET	VARCHAR2(20)	The client's subnet mask
NET_LATENCY_IN_MS	NUMBER	The client's HTTP response time with the server
NET_BANDWIDTH_IN_KBITPS	NUMBER	The client's download bandwidth from the server
BROWSER_TYPE	VARCHAR2(100)	The name of the browser used to run CSA
BROWSER_VERSION	VARCHAR2(20)	The version of the browser used to run CSA
BROWSER	VARCHAR2(121)	A summary column that combines the browser name and version
BROWSER_JVM_VENDOR	VARCHAR2(20)	The version of the JVM used to run the applet
BROWSER_JVM_VERSION	VARCHAR2(20)	The version of the JVM used to run the applet
BROWSER_PROXY_SERVER	VARCHAR2(4000)	The proxy server used by the browser
BROWSER_PROXY_EXCEPTIONS	VARCHAR2(4000)	The client's browser proxy exceptions
BROWSER_CACHE_SIZE_IN_MB	NUMBER	The client browser's disk cache size
BROWSER_CACHE_UPATE_FRQ	VARCAHR2(200)	The browser's cache update policy
BROWSER_HTTP1_1_SUPPORT	VARCHAR2(1)	Whether or not the browser supports HTTP 1.1
REFERRING_URL_HEADER	VARCHAR2(4000)	The URL from which the user came to CSA, minus the query string
REFERRING_URL_PARAMS	VARCHAR2(4000)	The query string of the URL from which the user came to CSA
REFURL	VARCHAR2(4000)	The complete URL from which the user came to CSA
CSA_URL_HEADER	VARCHAR2(4000)	The URL from which the user ran CSA, minus the query string
CSA_URL_PARAMS	VARCHAR2(4000)	The query string of the URL from which the user ran CSA
CSAURL	VARCHAR2(4000)	The complete URL from which the user ran CSA

**Table 6–122 (Cont.) MGMT\$CSA\_COLLECTIONS**

Column	Datatype	Description
DESTINAION_URL_HEADER	VARCHAR2(4000)	The destination URL, minus the query string
DESTINATION_URL_PARAMS	VARCHAR2(4000)	The query string of the destination URL
DESTURL	VARCHAR2(4000)	The complete destination URL
CONNECTION_TYPE	NUMBER	The estimated connection type, based on the download bandwidth (1=LAN, 2=cable, 3=dialup)
IS_WINDOWS_ADMIN	VARCHAR2(1)	Whether or not the os user is a windows admin
WINDOWS_DOMAIN	VARCHAR2(100)	The windows domain of the host
BROWSER_PROXY_ENABLED	VARCHAR2(1)	Whether or not the proxy server is enabled in the browser
AUTO_CONFIG_URL	VARCHAR2(4000)	The URL of the proxy auto-configuration script used by the browser
NUMBER_OF_COOKIES	NUMBER	The number of cookies collected by CSA
NUMBER_OF_CUSTOM_VALUES	NUMBER	The number of custom properties collected by CSA
HARDWARE	VARCHAR2(4000)	A summary of the system configuration, machine architecture, memory, disk space, and CPU
HARDWARE_VENDOR_NAME	VARCHAR2(128)	The name of the hardware vendor, e.g. iDell
SYSTEM_CONFIG	VARCHAR2(4000)	The client's system configuration
MACHINE_ARCHITECTURE	VARCHAR2(500)	The client's machine architecture
BUS_FREQ_IN_MHZ	NUMBER	The frequency of the motherboard's FSB
MEMORY_SIZE_IN_MB	NUMBER	The total amount of physical memory
AVAIL_MEMORY_SIZE_IN_MB	NUMBER	The amount of available physical memory when CSA was run
LOCAL_DISK_SPACE_IN_GB	NUMBER	The total amount of disk space
AVAIL_LOCAL_DISK_SPACE_IN_GB	NUMBER	The available disk space
CPU_COUNT	NUMBER(8)	The number of CPUs
SYSTEM_SERIAL_NUMBER	VARCAHR2(100)	The host's serial number
MIN_CPU_SPEED_IN_MHZ	NUMBER	The minimum possible CPU speed
MAX_CPU_SPEED_IN_MHZ	NUMBER	The maximum possible CPU speed
CPU	VARCHAR2(673)	The CPU vendor, implementation, and frequency
CPU_BOARD_COUNT	NUMBER(8)	The number of CPU boards
IOCARD_COUNT	NUMBER(8)	The number of IO cards
NIC_COUNT	NUMBER	The number of NICs
FAN_COUNT	NUMBER(8)	The number of fans
POWER_SUPPLY_COUNT	NUMBER(8)	The number of power supplies
SYSTEM_BIOS	VARCHAR2(100)	The system BIOS
OPERATINGSYSTEM	VARCHAR2(453)	A summary of the OS name, version, update level, address length, and distributor version
OS_NAME	VARCHAR2(128)	The OS name
OS_VENDOR_NAME	VARCHAR2(128)	The OS vendor name

**Table 6–122 (Cont.) MGMT\$CSA\_COLLECTIONS**

<b>Column</b>	<b>Datatype</b>	<b>Description</b>
OS_BASE_VERSION	VARCAR2(100)	The OS base version
OS_UPDATE_LEVEL	VARCHAR2(100)	The OS update level
OS_DISTRIBUTOR_VERSION	VARCHAR2(100)	The OS distributor version
MAX_SWAP_SPACE_IN_MB	NUMBER	The maximum amount of swap space
OS_ADDRESS_LENGTH_IN_BITS	VARCHAR2(20)	The OS address length in bits
MAX_PROCESS_VIRTUAL_MEMORY	NUMBER	The maximum amount of virtual memory that can be allocated to a process
TIMEZONE	VARCHAR2(64)	The time zone as reported in the registry
TIMEZONE_REGION	VARCHAR2(64)	The time zone region as reported by the JVM
TIMEZONE_DELTA	NUMBER	The offset in minutes from GMT
NUMBER_OF_OS_PROPERTIES	NUMBER	The number of OS properties found
NUMBER_OF_OS_PATCHES	NUMBER	The number of OS patches found
NUMBER_OF_OS_FILESYSTEMS	NUMBER	The number of filesystems found
NUMBER_OF_OS_REGISTERED_SW	NUMBER	The number of OS-registered software products found
SNAPSHOT_ID	RAW(16)	The GUID of this configuration
TARGET_ID	RAW(16)	The GUID of the collector target
INTERNAL_TARGET_NAME	VARCHAR2(264)	The internal name of the client configuration
INTERNAL_TARGET_TYPE	VARCHAR2(64)	oracle_csa_client
COLLECTION_DURATION	NUMBER(16)	The amount of time it took to run CSA
LOADED_TIMESTAMP	DATE	The time at which the data was loaded into the repository
APPLET_VERSION	VARCHAR2(20)	The version of the applet
TARGET_ID_METHOD	VARCHAR2(100)	not used
CUSTOM_CLASS	VARCHAR2(1000)	The name of the custom class (if any)
CUSTOM_CLASS_VERSION	VARCHAR2(1000)	not used
KEY1	VARCHAR2(4000)	Custom key 1(optional)
KEY2	VARCHAR2(4000)	Custom key 2 (optional)
KEY3	VARCHAR2(4000)	Custom key 3 (optional)
PROXY_TARGET_NAME	VARCHAR2(256)	The name of the collector target
PROXY_TARGET_DISPLAY_NAME	VARCHAR2(256)	The display name of the collector target
PROXY_TARGET_ID	RAW(16)	The GUID of the collector target
RULES_COUNT	NUMBER	The total number of rules evaluated (including rules with status of NA)
RULES_NA_COUNT	NUMBER	The number of rules that were not applicable
RULES_PASSED_COUNT	NUMBER	The number of rules that passed
RULES_INFO_COUNT	NUMBER	The number of rules that failed with status info
RULES_WARNING_COUNT	NUMBER	The number of rules that failed with status warning

**Table 6–122 (Cont.) MGMT\$CSA\_COLLECTIONS**

Column	Datatype	Description
RULES_CRITICAL_COUNT	NUMBER	The number of rules that failed with status critical

## 6.14.2 MGMT\$CSA\_FAILED

MGMT\$CSA\_FAILED displays all failed collections.

**Table 6–123 MGMT\$CSA\_FAILED**

Column	Datatype	Description
ID	RAW(16)	The GUID of this failed collection
TIMESTAMP	DATE	The time at which this failed collection occurred
TIMEZONE_DELTA	NUMBER	The offset in minutes from GMT
SAVED_TIMESTAMP	DATE	The time at which the data was loaded in the repository
EFFECTIVE_IP	VARCHAR2(20)	The effective IP address of the client
APPID	VARCHAR2(128)	The collection tag
REFERRING_URL_HEADER	VARCHAR2(4000)	The URL from which the user was referred to CSA, minus the query string
REFERRING_URL_PARAMS	VARCHAR2(4000)	The query string of the URL from which the user was referred to CSA
CSA_URL_HEADER	VARCHAR2(4000)	The URL from which the user ran CSA, minus the query string
CSA_URL_PARAMS	VARCHAR2(4000)	The query string of the URL from which the user tried to run CSA
DESTINATION_URL_HEADER	VARCHAR2(4000)	The destination URL minus the query string
DESTINATION_URL_PARAMS	VARCHAR2(4000)	The query string of the destination URL
BROWSER_TYPE	VARCHAR2(100)	The type of browser used to run CSA
BROWSER_VERSION	VARCHAR2(20)	The version of the browser used to run CSA
BROWSER_JVM_VENDOR	VARCHAR(100)	The vendor of the JVM used to run CSA
BROWSER_JVM_VERSION	VARCHAR2(20)	The version of the JVM used to run CSA
OS_ARCH	VARCHAR2(100)	The OS architecture of the client as reported in the ios.arch Java system property
OS_NAME	VARCHAR2(100)	The OS name of the client as reported in the ios.name Java system property
HTTP_REQUEST_USER_AGENT	VARCHAR2(100)	The HTTP user-agent header sent by the client
ERROR_CODE	VARCHAR2(1)	The error condition that caused the failed collection (0=OS not supported, 1=browser not supported, 2=applet certificate refused by user, 3=other error)
ERROR_TEXT	VARCHAR2(1024)	Text that is collected along with the error code, such as a stack trace

## 6.14.3 MGMT\$CSA\_HOST\_OS\_COMPONENTS

MGMT\$CSA\_HOST\_OS\_COMPONENTS displays all OS components found on CSA client machines.

**Table 6–124 MGMT\$CSA\_HOST\_OS\_COMPONENTS**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
TYPE	VARCHAR2(100)	The type of the component
NAME	VARCHAR2(128)	The name of the component
VERSION	VARCHAR2(100)	The version of the component
DESCRIPTION	VARCHAR2(2000)	The description of the component
ISNTALLATION_DATE	DATE	The date the component was installed
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
TARGET_ID	RAW(16)	The target GUID of the collector target
COLLECTION_TIMESTAMP	DATE	The time at which the client configuration was collected

#### 6.14.4 MGMT\$CSA\_HOST\_SW

MGMT\$CSA\_HOST\_SW displays all OS-registered software found on CSA hosts.

**Table 6–125 MGMT\$CSA\_HOST\_SW**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
NAME	VARCHAR2(128)	The name of the software
VENDOR_NAME	VARCHAR2(128)	The name of the software vendor
VERSION	VARCHAR2(100)	The version of the software
INSTALLATION_DATE	DATE	The date on which the software as installed
INSTALLED_LOCATION	VARCHAR2(1024)	The location in which the software is installed
DESCRIPTION	VARCHAR2(2000)	The description of the software
VENDOR_SW_SPECIFIC_INFO	VARCHAR2(4000)	Any additional information provided by the vendor
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
TARGET_ID	RAW(16)	The ID of the collector target
COLLECTION_TIMESTAMP	DATE	The time at which the data was collected

#### 6.14.5 MGMT\$CSA\_HOST\_COOKIES

MGMT\$CSA\_HOST\_COOKIES displays the cookies collected with client configurations.

**Table 6–126 MGMT\$CSA\_HOST\_COOKIES**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
NAME	VARCHAR2(4000)	The name of the cookie
VALUE	VARCHAR2(4000)	The payload of the cookie
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
TARGET_ID	RAW(16)	The ID of the collector target
COLLECTION_TIMESTAMP	DATE	The time at which the data was collected

## 6.14.6 MGMT\$CSA\_HOST\_CUSTOM

MGMT\$CSA\_HOST\_CUSTOM displays the custom properties collected with client configurations.

**Table 6–127 MGMT\$CSA\_HOST\_CUSTOM**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
TYPE	VARCHAR2(512)	The category of the custom property
NAME	VARCHAR2(512)	The name of the custom property
TYPE_UI	VARCHAR2(4000)	The display category of the custom property
NAME_UI	VARCHAR2(4000)	The display name of the custom property
VALUE	VARCHAR2(4000)	The value of the custom property
DISPLAY_UI	VARCHAR2(1)	Should this property be displayed in the UI? Y or N
HISTORY_TRACKING	VARCHAR2(1)	Not used
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
TARGET_ID	RAW(16)	The ID of the collector target
COLLECTION_TIMESTAMP	DATE	The time at which the data was collected

## 6.14.7 MGMT\$CSA\_HOST\_RULES

MGMT\$CSA\_HOST\_RULES displays the rules that were evaluated with each client configuration.

**Table 6–128 MGMT\$CSA\_HOST\_RULES**

Column	Datatype	Description
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
NAME	VARCHAR2(128)	The name of the rule
DESCRIPTION	VARCHAR2(256)	The description of the rule
STATUS	NUMBER	The status of the rule (-2=NA, 15=passed, 18=info, 20=warning, 25=critical)
MOREINFO	VARCHAR2(1024)	Any additional information for the rule

## 6.14.8 MGMT\$CSA\_HOST\_CPUS

MGMT\$CSA\_HOST\_CPUS displays information about the CPUs of CSA hosts. CSA assumes that in a multi-CPU host, all CPUs are identical.

**Table 6–129 MGMT\$CSA\_HOST\_CPUS**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
VENDOR_NAME	VARCHAR2(128)	The name of the CPU vendor
FREQ_IN_MHZ	NUMBER	The clock frequency of the CPU
ECACHE_IN_MB	NUMBER	The size of the extended cache
IMPL	VARCAHR2(500)	The CPU implementation
REVISION	VARCHAR2(2000)	The CPU revision
MASK	VARCHAR2(500)	The CPU mask

**Table 6–129 (Cont.) MGMT\$CSA\_HOST\_CPUS**

Column	Datatype	Description
NUMBER_OF_CPUS	NUMBER	The number of CPUs
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
TARGET_ID	RAW(16)	The ID of the collector target
COLLECTION_TIMESTAMP	DATE	The time at which the data was collected

### 6.14.9 MGMT\$CSA\_HOST\_IOCARDS

MGMT\$CSA\_HOST\_IOCARDS displays all IO cards collected from client configurations.

**Table 6–130 MGMT\$CSA\_HOST\_IOCARDS**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
VENDOR_NAME	VARCHAR2(128)	The name of the IO card vendor
NAME	VARCHAR2(128)	The name of the IO card
FREQ_IN_MHZ	NUMBER	The frequency of the IO card bus
BUS	VARCHAR2(500)	The bus type (PCI or AGP)
REVISION	VARCHAR2(2000)	The IO card revision
NUMBER_OF_IOCARDS	NUMBER	The number of cards
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
TARGET_ID	RAW(16)	The ID of the collector target
COLLECTION_TIMESTAMP	DATE	The time at which the data was collected

### 6.14.10 MGMT\$CSA\_HOST\_NICS

MGMT\$CSA\_HOST\_NICS displays all network interface cards collected from client configurations.

**Table 6–131 MGMT\$CSA\_HOST\_NICS**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
NAME	VARCHAR2(128)	The name of the NIC
DESCRIPTION	VARCHAR2(500)	The description of the NIC
FLAGS	VARCHAR2(1024)	Flags set on the NIC – not applicable for Windows
MAX_TRANSFER_UNIT	NUMBER	The maximum transfer unit of the NIC
INET_ADDRESS	VARCHAR2(20)	The IP address of the NIC
MASK	VARCHAR2(20)	The subnet mask of the NIC
BROADCAST_ADDRESS	VARCHAR2(20)	The broadcast address of the NIC
MAC_ADDRESS	VARCHAR2(20)	The MAC address of the NIC
HOSTNAME_ALIASES	VARCHAR2(4000)	Any aliases for the hostname that are stored in the NIC
DEFAULT_GATEWAY	VARCHAR2(20)	The default gateway for the NIC
DHCP_ENABLED	VARCHAR2(1)	Whether or not DHCP is enabled



**Table 6–131 (Cont.) MGMT\$CSA\_HOST\_NICS**

Column	Datatype	Description
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
TARGET_ID	RAW(16)	The ID of the collector target
COLLECTION_TIMESTAMP	DATE	The time at which the data was collected

### 6.14.11 MGMT\$CSA\_HOST\_OS\_PROPERTIES

MGMT\$CSA\_HOST\_OS\_PROPERTIES displays all OS properties, such as environment variables, found on CSA hosts.

**Table 6–132 MGMT\$CSA\_HOST\_OS\_PROPERTIES**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
SOURCE	VARHCAR2(128)	The source (e.g. the system environment) of the property
NAME	VARCHAR2(128)	The name of the property
VALUE	VARCHAR2(2000)	The value of the property
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
TARGET_ID	RAW(16)	The ID of the collector target
COLLECTION_TIMESTAMP	DATE	The time at which the data was collected

### 6.14.12 MGMT\$CSA\_HOST\_OS\_FILESYSTEMS

MGMT\$CSA\_HOST\_OS\_FILESYSTEMS displays all filesystems found on CSA hosts.

**Table 6–133 MGMT\$CSA\_HOST\_OS\_FILESYSTEMS**

Column	Datatype	Description
DISPLAY_TARGET_NAME	VARCHAR2(256)	The display name of the client
RESOURCE_NAME	VARCHAR2(128)	The name of the filesystem
MOUNT_LOCATION	VARCHAR2(1024)	The location from which it is mounted
TYPE	VARCHAR2(100)	The filesystem type
DISK_SPACE_IN_GB	NUMBER	The total disk space
AVAIL_DISK_SPACE_IN_GB	NUMBER	The available disk space
LOCAL_DRIVE	VARCHAR2(1)	The Windows drive letter on which it is mounted
MOUNT_OPTIONS	VARCHAR2(1024)	The mount options
SNAPSHOT_ID	RAW(16)	The snapshot ID of the client configuration
TARGET_ID	RAW(16)	The ID of the collector target
COLLECTION_TIMESTAMP	DATE	The time at which the data was collected

### 6.14.13 MGMT\$ECM\_CONFIG\_HISTORY

MGMT\$ECM\_CONFIG\_HISTORY displays the data needed for generic categories.

**Table 6–134 MGMT\$ECM\_CONFIG\_HISTORY**

Column	Datatype	Description
DELTATIME	DATE	The time at which the diff was recorded
DELTAGUID	RAW(16)	The unique ID of the diff
TIMEZONE	VARCHAR2(64)	The time zone of the target
TARGET_NAME	VARCHAR2(256)	The name of the target
HOSTNAME	VARCHAR2(256)	The name of the target's host
TARGET_TYPE	VARCHAR2(64)	The type of the target
SNAPSHOTTYPE	VARCHAR2(64)	The type of the snapshot in which the diff was recorded
COLLECTIONTYPE	VARCHAR2(64)	The name of the table in which the diff occurred
TABLE_PATH	VARCHAR2(1000)	The full path of the table
CATEGORY	VARCHAR2(1132)	The fully qualified leaf category
OPERATION	VARCHAR2(10)	The type of change (update, delete, insert, same)
ROWGUID	RAW(16)	The entry ID GUID
ATTRIBUTE	VARCHAR2(64)	Attribute column name
NEWVALUE	VARCHAR2(4000)	New value of attribute
OLDVALUE	VARCHAR2(4000)	Old value of attribute

#### 6.14.14 MGMT\$ECM\_CONFIG\_HISTORY\_KEY1

MGMT\$ECM\_CONFIG\_HISTORY\_KEY1 displays the data needed for specific categories.

**Table 6–135 MGMT\$ECM\_CONFIG\_HISTORY\_KEY1**

Column	Datatype	Description
DELTATIME	DATE	The time at which the diff was recorded
DELTAGUID	RAW(16)	The unique ID of the diff
TIMEZONE	VARCHAR2(64)	The time zone of the target
TARGET_NAME	VARCHAR2(256)	The name of the target
HOSTNAME	VARCHAR2(256)	The name of the target's host
TARGET_TYPE	VARCHAR2(64)	The type of the target
SNAPSHOTTYPE	VARCHAR2(64)	The type of the snapshot in which the diff was recorded
COLLECTIONTYPE	VARCHAR2(64)	The name of the table in which the diff occurred
TABLE_PATH	VARCHAR2(1000)	The full path of the table
CATEGORY	VARCHAR2(1132)	The fully qualified leaf category
OPERATION	VARCHAR2(10)	The type of change (update, delete, insert, same)
KEY1	VARCHAR2(4000)	Value of Key 1
ATTRIBUTE	VARCHAR2(64)	Attribute column name
NEWVALUE	VARCHAR2(4000)	New value of attribute
OLDVALUE	VARCHAR2(4000)	Old value of attribute

### 6.14.15 MGMT\$ECM\_CONFIG\_HISTORY\_KEY2

MGMT\$ECM\_CONFIG\_HISTORY\_KEY2 displays the data needed for specific categories.

**Table 6–136 MGMT\$ECM\_CONFIG\_HISTORY\_KEY2**

Column	Datatype	Description
DELTATIME	DATE	The time at which the diff was recorded
DELTAGUID	RAW(16)	The unique ID of the diff
TIMEZONE	VARCHAR2(64)	The time zone of the target
TARGET_NAME	VARCHAR2(256)	The name of the target
HOSTNAME	VARCHAR2(256)	The name of the target's host
TARGET_TYPE	VARCHAR2(64)	The type of the target
SNAPSHOTTYPE	VARCHAR2(64)	The type of the snapshot in which the diff was recorded
COLLECTIONTYPE	VARCHAR2(64)	The name of the table in which the diff occurred
TABLE_PATH	VARCHAR2(1000)	The full path of the table
CATEGORY	VARCHAR2(1132)	The fully qualified leaf category
OPERATION	VARCHAR2(10)	The type of change (update, delete, insert, same)
KEY1	VARCHAR2(4000)	Value of Key 1
KEY2	VARCHAR2(4000)	Value of Key 2
ATTRIBUTE	VARCHAR2(64)	Attribute column name
NEWVALUE	VARCHAR2(4000)	New value of attribute
OLDVALUE	VARCHAR2(4000)	Old value of attribute

### 6.14.16 MGMT\$ECM\_CONFIG\_HISTORY\_KEY3

MGMT\$ECM\_CONFIG\_HISTORY\_KEY3 displays the data needed for specific categories.

**Table 6–137 MGMT\$ECM\_CONFIG\_HISTORY\_KEY3**

Column	Datatype	Description
DELTATIME	DATE	The time at which the diff was recorded
DELTAGUID	RAW(16)	The unique ID of the diff
TIMEZONE	VARCHAR2(64)	The time zone of the target
TARGET_NAME	VARCHAR2(256)	The name of the target
HOSTNAME	VARCHAR2(256)	The name of the target's host
TARGET_TYPE	VARCHAR2(64)	The type of the target
SNAPSHOTTYPE	VARCHAR2(64)	The type of the snapshot in which the diff was recorded
COLLECTIONTYPE	VARCHAR2(64)	The name of the table in which the diff occurred
TABLE_PATH	VARCHAR2(1000)	The full path of the table
CATEGORY	VARCHAR2(1132)	The fully qualified leaf category
OPERATION	VARCHAR2(10)	The type of change (update, delete, insert, same)
KEY1	VARCHAR2(4000)	Value of Key 1
KEY2	VARCHAR2(4000)	Value of Key 2

**Table 6–137 (Cont.) MGMT\$ECM\_CONFIG\_HISTORY\_KEY3**

Column	Datatype	Description
KEY3	VARCHAR2(4000)	Value of Key 3
ATTRIBUTE	VARCHAR2(64)	Attribute column name
NEWVALUE	VARCHAR2(4000)	New value of attribute
OLDVALUE	VARCHAR2(4000)	Old value of attribute

### 6.14.17 MGMT\$ECM\_CONFIG\_HISTORY\_KEY4

MGMT\$ECM\_CONFIG\_HISTORY\_KEY4 displays the data needed for specific categories.

**Table 6–138 MGMT\$ECM\_CONFIG\_HISTORY\_KEY4**

Column	Datatype	Description
DELTATIME	DATE	The time at which the diff was recorded
DELTAID	RAW(16)	The unique ID of the diff
TIMEZONE	VARCHAR2(64)	The time zone of the target
TARGET_NAME	VARCHAR2(256)	The name of the target
HOSTNAME	VARCHAR2(256)	The name of the target's host
TARGET_TYPE	VARCHAR2(64)	The type of the target
SNAPSHOTTYPE	VARCHAR2(64)	The type of the snapshot in which the diff was recorded
COLLECTIONTYPE	VARCHAR2(64)	The name of the table in which the diff occurred
TABLE_PATH	VARCHAR2(1000)	The full path of the table
CATEGORY	VARCHAR2(1132)	The fully qualified leaf category
OPERATION	VARCHAR2(10)	The type of change (update, delete, insert, same)
KEY1	VARCHAR2(4000)	Value of Key 1
KEY2	VARCHAR2(4000)	Value of Key 2
KEY3	VARCHAR2(4000)	Value of Key 3
KEY4	VARCHAR2(4000)	Value of Key 4
ATTRIBUTE	VARCHAR2(64)	Attribute column name
NEWVALUE	VARCHAR2(4000)	New value of attribute
OLDVALUE	VARCHAR2(4000)	Old value of attribute

### 6.14.18 MGMT\$ECM\_CONFIG\_HISTORY\_KEY5

MGMT\$ECM\_CONFIG\_HISTORY\_KEY5 displays the data needed for specific categories.

**Table 6–139 MGMT\$ECM\_CONFIG\_HISTORY\_KEY5**

Column	Datatype	Description
DELTATIME	DATE	The time at which the diff was recorded
DELTAID	RAW(16)	The unique ID of the diff
TIMEZONE	VARCHAR2(64)	The time zone of the target
TARGET_NAME	VARCHAR2(256)	The name of the target

**Table 6–139 (Cont.) MGMT\$ECM\_CONFIG\_HISTORY\_KEY5**

Column	Datatype	Description
HOSTNAME	VARCHAR2(256)	The name of the target's host
TARGET_TYPE	VARCHAR2(64)	The type of the target
SNAPSHOTTYPE	VARCHAR2(64)	The type of the snapshot in which the diff was recorded
COLLECTIONTYPE	VARCHAR2(64)	The name of the table in which the diff occurred
TABLE_PATH	VARCHAR2(1000)	The full path of the table
CATEGORY	VARCHAR2(1132)	The fully qualified leaf category
OPERATION	VARCHAR2(10)	The type of change (update, delete, insert, same)
KEY1	VARCHAR2(4000)	Value of Key 1
KEY2	VARCHAR2(4000)	Value of Key 2
KEY3	VARCHAR2(4000)	Value of Key 3
KEY4	VARCHAR2(4000)	Value of Key 4
KEY5	VARCHAR2(4000)	Value of Key 5
ATTRIBUTE	VARCHAR2(64)	Attribute column name
NEWVALUE	VARCHAR2(4000)	New value of attribute
OLDVALUE	VARCHAR2(4000)	Old value of attribute

### 6.14.19 MGMT\$ECM\_CONFIG\_HISTORY\_KEY6

MGMT\$ECM\_CONFIG\_HISTORY\_KEY6 displays the data needed for specific categories.

**Table 6–140 MGMT\$ECM\_CONFIG\_HISTORY\_KEY6**

Column	Datatype	Description
DELTA TIME	DATE	The time at which the diff was recorded
DELTA GUID	RAW(16)	The unique ID of the diff
TIMEZONE	VARCHAR2(64)	The time zone of the target
TARGET_NAME	VARCHAR2(256)	The name of the target
HOSTNAME	VARCHAR2(256)	The name of the target's host
TARGET_TYPE	VARCHAR2(64)	The type of the target
SNAPSHOTTYPE	VARCHAR2(64)	The type of the snapshot in which the diff was recorded
COLLECTIONTYPE	VARCHAR2(64)	The name of the table in which the diff occurred
TABLE_PATH	VARCHAR2(1000)	The full path of the table
CATEGORY	VARCHAR2(1132)	The fully qualified leaf category
OPERATION	VARCHAR2(10)	The type of change (update, delete, insert, same)
KEY1	VARCHAR2(4000)	Value of Key 1
KEY2	VARCHAR2(4000)	Value of Key 2
KEY3	VARCHAR2(4000)	Value of Key 3
KEY4	VARCHAR2(4000)	Value of Key 4
KEY5	VARCHAR2(4000)	Value of Key 5
KEY6	VARCHAR2(4000)	Value of Key 6

**Table 6–140 (Cont.) MGMT\$ECM\_CONFIG\_HISTORY\_KEY6**

Column	Datatype	Description
ATTRIBUTE	VARCHAR2(64)	Attribute column name
NEWVALUE	VARCHAR2(4000)	New value of attribute
OLDVALUE	VARCHAR2(4000)	Old value of attribute

### 6.14.20 MGMT\$HW\_NIC

MGMT\$HW\_NIC displays performance information for host hardware network cards.

**Table 6–141 MGMT\$HW\_NIC**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	The host name
NAME	VARCHAR2(128)	The name of the card
INET_ADDRESS	VARCHAR2(20)	The IP address of the card
MAX_TRANSFER_UNIT	NUMBER	The maximum transfer unit
BROADCAST_ADDRESS	VARCHAR2(20)	The broadcast address of the card
MASK	VARCHAR2(20)	The card's subnet mask
FLAGS	VARCHAR2(1024)	The flags on the card
MAC_ADDRESS	VARCHAR2(20)	The MAC address of the card
HOST_ALIASES	VARCHAR2(4000)	The aliases for the hostname
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.21 MGMT\$OS\_COMPONENTS

MGMT\$OS\_COMPONENTS displays performance information for host OS components.

**Table 6–142 MGMT\$OS\_COMPONENTS**

Column	Datatype	Description
HOST	VARCHAR2(256)	The name of the host
NAME	VARCHAR2(128)	The name of the component
TYPE	VARCHAR2(100)	The type of the component
VERSION	VARCHAR2(100)	The version of the component
DESCRIPTION	VARCHAR2(2000)	The description of the component
INSTALLATION_DATE	DATE	The installation date of the component
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.22 MGMT\$OS\_FS\_MOUNT

MGMT\$OS\_FS\_MOUNT displays performance information for mounted filesystems.

**Table 6–143 MGMT\$OS\_FS\_MOUNT**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	The host name
RESOURCE_NAME	VARCHAR2(128)	The name of the mounted resource

**Table 6–143 (Cont.) MGMT\$OS\_FS\_MOUNT**

Column	Datatype	Description
TYPE	VARCHAR2(100)	The filesystem type
MOUNT_LOCATION	VARCHAR2(1024)	The mount location
MOUNT_OPTIONS	VARCHAR2(1024)	The mount options
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.23 MGMT\$OS\_HW\_SUMMARY

MGMT\$OS\_HW\_SUMMARY displays summary information for both operating systems and hardware.

**Table 6–144 MGMT\$OS\_HW\_SUMMARY**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	The host name
DOMAIN	VARCHAR2(500)	The host's domain
OS_SUMMARY	VARCHAR2(352)	A summary of OS information
SYSTEM_CONFIG	VARCHAR2(4000)	The system configuration
MA	VARCHAR2(500)	The machine architecture
FREQ	NUMBER	The CPU frequency
MEM	NUMBER	The total amount of memory
DISK	NUMBER	The total amount of disk space
CPU_COUNT	NUMBER	The number of CPUs
VENDOR_NAME	VARCHAR2(128)	The name of the system vendor
OS_VENDOR	VARCHAR2(128)	The name of the OS vendor
DISTRIBUTOR_VERSION	VARCHAR2(100)	The OS distributor version
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.24 MGMT\$OS\_KERNEL\_PARAMS

MGMT\$OS\_KERNEL\_PARAMS displays performance information for OS kernel parameters.

**Table 6–145 MGMT\$OS\_KERNEL\_PARAMS**

Column	Datatype	Description
HOST	VARCHAR2(256)	The name of the host
SOURCE	VARCHAR2(128)	The source of the parameter
NAME	VARCHAR2(128)	The name of the parameter
VALUE	VARCHAR2(2000)	The value of the parameter
DATATYPE	VARCHAR2(4000)	The data type of the parameter
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.25 MGMT\$OS\_PATCHES

MGMT\$OS\_PATCHES displays performance information for OS patches.

**Table 6–146** *MGMT\$OS\_PATCHES*

Column	Datatype	Description
HOST	VARCHAR2(256)	The name of the host
OS_EXTENDED	VARCHAR2(352)	A summary column that includes update level and address length
OS	VARCHAR2(229)	A summary column that includes name and version
PATCH	VARCHAR2(128)	The name of the patch

### 6.14.26 MGMT\$OS\_SUMMARY

MGMT\$OS\_SUMMARY displays operating system summary information.

**Table 6–147** *MGMT\$OS\_SUMMARY*

Column	Datatype	Description
HOST	VARCHAR2(256)	The name of the host
NAME	VARCHAR2(128)	The name of the OS
VENDOR_NAME	VARCHAR2(128)	The name of the OS vendor
BASE_VERSION	VARCHAR2(100)	The base version of the OS
UPDATE_LEVEL	VARCHAR2(100)	The update level of the OS
DISTRIBUTOR_VERSION	VARCHAR2(100)	The distributor version of the OS
MAX_SWAP_SPACE_IN_MB	NUMBER	The maximum amount of swap space
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.27 MGMT\$SOFTWARE\_COMP\_PATCHSET

MGMT\$SOFTWARE\_COMP\_PATCHSET displays information on components and patchsets.

**Table 6–148** *MGMT\$SOFTWARE\_COMP\_PATCHSET*

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	The name of the host
HOME_NAME	VARCHAR2(64)	The name of the Oracle Home
HOME_LOCATION	VARCHAR2(128)	The path of the Oracle Home
COMPONENT_NAME	VARCHAR2(128)	The name of the component
COMPONENT_BASE_VERSION	VARCHAR2(64)	The base version of the component
COMPONENT_VERSION	VARCHAR2(64)	The current version of the component
PATCHSET_NAME	VARCHAR2(128)	The name of the patchset
PATCHSET_VERSION	VARCHAR2(64)	The version of the patchset
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.28 MGMT\$SOFTWARE\_COMPONENT\_ONEOFF

MGMT\$SOFTWARE\_COMPONENT\_ONEOFF displays information on oneoff patches.



**Table 6–149 MGMT\$SOFTWARE\_COMPONENT\_ONEOFF**

Column	Datatype	Description
HOST_NAME	VARCHAR2(256)	The name of the host
HOME_NAME	VARCHAR2(64)	The name of the Oracle Home
HOME_LOCATION	VARCHAR2(128)	The path of the Oracle Home
COMPONENT_NAME	VARCHAR2(128)	The name of the component
COMPONENT_EXTERNAL_NAME	VARCHAR2(128)	The external name of the component
COMPONENT_BASE_VERSION	VARCHAR2(64)	The base version of the component
COMPONENT_VERSION	VARCHAR2(64)	The current version of the component
PATCH_ID	VARCHAR2(128)	The ID of the patch
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.29 MGMT\$SOFTWARE\_COMPONENTS

MGMT\$SOFTWARE\_COMPONENTS displays information on components.

**Table 6–150 MGMT\$SOFTWARE\_COMPONENTS**

Column	Datatype	Description
NAME	VARCHAR2(128)	The name of the component
EXTERNAL_NAME	VARCHAR2(128)	The external name of the component
BASE_VERSION	VARCHAR2(64)	The base version of the component
PATCHSETS_IN_HOME	VARCHAR2(4000)	List of patchsets in home
VERSION	VARCHAR2(64)	The version of the component
HOST_NAME	VARCHAR2(256)	The name of the host
HOME_LOCATION	VARCHAR2(128)	The path of the Oracle Home
HOME_NAME	VARCHAR2(64)	The name of the Oracle Home
DESCRIPTION	VARCHAR2(1024)	A description of the component
INSTALLER_VERSION	VARCHAR2(64)	The version of the installer
MIN_DEINSTALLER_VERSION	VARCHAR2(64)	The minimum deinstaller version
INSTALL_TIMESTAMP	DATE	The installation date
IS_TOP_LEVEL	VARCHAR2(1)	Whether or not the component is a top-level component
INTERIM_PATCHES_IN_HOME	VARCHAR2(4000)	The interim patches in the home
BUGS_FIXED_BY_INTERM_INPATCEHS	VARCHAR2(4000)	A list of bugs fixed by the interim patches in the home
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.30 MGMT\$SOFTWARE\_DEPENDENCIES

MGMT\$SOFTWARE\_COMPONENTS displays information on components.

**Table 6–151 MGMT\$SOFTWARE\_DEPENDENCIES**

Column	Datatype	Description
HOST_NAME	VARHCAR2(256)	The name of the host

**Table 6–151 (Cont.) MGMT\$SOFTWARE\_DEPENDENCIES**

Column	Datatype	Description
REFERENCER_HOME_NAME	VARCHAR2(64)	The Oracle Home of the referencing component
REFERENCER_HOME_LOCATION	VARCHAR2(128)	The Oracle Home path of the referencing component
REFERENCER_NAME	VARCHAR2(128)	The name of the referencing component
REFERENCER_BASE_VERSION	VARCHAR2(64)	The base version of the referencing component
REFERENCED_HOME_NAME	VARCHAR2(64)	The Oracle Home of the referenced component
REFERENCED_HOME_LOCATION	VARCHAR2(128)	The Oracle Home path of the referenced component
REFERENCED_NAME	VARCHAR2(128)	The name of the referenced component
REFERENCED_BASE_VERSION	VARCHAR2(64)	The base version of the referenced component
SNAPSHOT_GUID	RAW(16)	The GUID of the configuration snapshot

### 6.14.31 MGMT\$SOFTWARE\_HOMES

MGMT\$SOFTWARE\_HOMES displays information about Oracle Homes.

**Table 6–152 MGMT\$SOFTWARE\_HOMES**

Column	Datatype	Description
HOST_NAME	VARHCAR2(256)	The name of the host
HOME_NAME	VARCHAR2(64)	The name of the Oracle Home
HOME_TYPE	VARCHAR2(11)	The type of the Oracle Home
HOME_LOCATION	VARCHAR2(128)	The path of the Oracle Home
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.32 MGMT\$SOFTWARE\_ONEOFF\_PATCHES

MGMT\$SOFTWARE\_ONEOFF\_PATCHES displays information on oneoff patches applied in Oracle Homes.

**Table 6–153 MGMT\$SOFTWARE\_ONEOFF\_PATCHES**

Column	Datatype	Description
HOST_NAME	VARHCAR2(256)	The name of the host
HOME_NAME	VARCHAR2(64)	The name of the Oracle Home
HOME_LOCATION	VARCHAR2(128)	The path of the Oracle Home
PATCH_ID	VARCHAR2(128)	The ID of the patch
INSTALL_TIMESTAMP	DATE	The installation date
DESCRIPTION	VARCHAR2(1024)	A description of the patch
IS_ROLLBACKABLE	VARCHAR2(1)	Whether or not the patch can be rolled back
SNAPSHOT_GUID	RAW(16)	The GUID of the configuration snapshot

### 6.14.33 MGMT\$SOFTWARE\_OTHERS

MGMT\$SOFTWARE\_OTHERS displays information on other software installed on hosts.

**Table 6–154 MGMT\$SOFTWARE\_OTHERS**

Column	Datatype	Description
HOST_NAME	VARHRCAR2(256)	The name of the host
SOFTWARE_NAME	VARCHAR2(128)	The name of the software
SOFTWARE_VENDOR	VARCHAR2(128)	The software vendor
SOFTWARE_VERSION	VARCHAR2(128)	The software version
INSTALLATION_DATE	DATE	The software installation date
INSTALLATION_LOCATION	VARCHAR2(1024)	The software installation location
SNAPSHOT_GUID	RAW(16)	The GUID of the configuration snapshot

### 6.14.34 MGMT\$SOFTWARE\_PATCHES\_IN\_HOMES

MGMT\$SOFTWARE\_PATCHES\_IN\_HOMES displays information on software patches in Oracle Homes.

**Table 6–155 MGMT\$SOFTWARE\_PATCHES\_IN\_HOMES**

Column	Datatype	Description
HOST_NAME	VARHRCAR2(256)	The name of the host
HOME_NAME	VARCHAR2(64)	The name of the Oracle Home
HOME_LOCATION	VARCHAR2(128)	The path of the Oracle Home
PATCH_ID	VARCHAR2(128)	The ID of the patch
BUGS_FIXED	VARCHAR2(4000)	List of bugs fixed by the patch
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

### 6.14.35 MGMT\$SOFTWARE\_PATCHSETS

MGMT\$SOFTWARE\_PATCHSETS displays information on patchsets installed in Oracle Homes.

**Table 6–156 MGMT\$SOFTWARE\_PATCHSETS**

Column	Datatype	Description
HOST_NAME	VARHRCAR2(256)	The name of the host
HOME_NAME	VARCHAR2(64)	The name of the Oracle Home
HOME_LOCATION	VARCHAR2(128)	The path of the Oracle Home
NAME	VARCHAR2(128)	The name of the patchset
VERSION	VARCHAR2(64)	The version of the patchset
DESCRIPTION	VARCHAR2(1024)	A description of the patchset
INSTALLER_VERSION	VARCHAR2(64)	The version of the installer
MIN_DEINSTALLER_VERSION	VARCHAR2(64)	The minimum deinstaller version
INSTALL_TIMESTAMP	DATE	The time at which the patchset was installed
SNAPSHOT_GUID	RAW(16)	The GUID of the config. snapshot

## 6.15 Database Cluster Views

## 6.15.1 MGMT\$CLUSTER\_INTERCONNECTS

MGMT\$CLUSTER\_INTERCONNECTS displays statistics of network interfaces on the hosts in clusters.

**Table 6–157 MGMT\$CLUSTER\_INTERCONNECTS**

Column	Datatype	Description
CLUSTER_NAME	VARCHAR2(256)	Cluster target name
HOST_NAME	VARCHAR2(256)	Host target status
IF_NAME	VARCHAR2(256)	Interface name
IF_SUBNET	VARCHAR2(16)	Subnet of the interface
IF_PUBLIC	VARCHAR2(10)	Whether the interface is public (YES) or private(NO)
TOTRATE_5MIN	NUMBER	The average total transfer rate (MB/sec) on the interface in the past 5 minutes
TOTERR_5MIN	NUMBER	The percentage of error packets on the interface in the past 5 minutes
INRATE_5MIN	NUMBER	The average input rate (MB/sec) on the interface in the past 5 minutes
CURR_WARNING	NUMBER	The number of warning alerts on the interface
CURR_CRITICAL	NUMBER	The number of critical alerts on the interface.
LATEST_COLLECTION_TIMESTAMP	DATE	The time when the data is collected.

## 6.15.2 MGMT\$RACDB\_INTERCONNECTS

MGMT\$RACDB\_INTERCONNECTS displays statistics of the inter-instance traffic of cluster databases.

**Table 6–158 MGMT\$RACDB\_INTERCONNECTS**

Column	Datatype	Description
CLUSTER_NAME	VARCHAR2(1024)	Cluster target name
DB_TARGET	VARCHAR2(256)	Cluster database target name
INSTANCE_TARGET	NUMBER	Cluster database instance target name.
INSTANCE_STATUS	NUMBER	Cluster database instance status
DB_NAME	VARCHAR2(1024)	Database name
SID	VARCHAR2(1024)	Instance name
IF_NAME	VARCHAR2(50)	The network interface the instance is using for inter-instance communication
HOST_NAME	VARCHAR2(256)	Host target name on which the instance is running
IF_IP	VARCHAR2(16)	The ip address of the network interface
IF_PUBLIC	VARCHAR2(10)	Whether the interface is public (YES) or private (NO)
IF_SOURCE	VARCHAR2(100)	Where does the instance pick up the interface
XFERRATE_5MIN	NUMBER	The transfer rate of the instance to other instances
CURR_WARNING	NUMBER	The number of warning alerts of the instance regarding inter_instance communication

**Table 6–158 (Cont.) MGMT\$RACDB\_INTERCONNECTS**

Column	Datatype	Description
CURR_CRITICAL	NUMBER	The number of critical alerts of the instance regarding inter_instance communication
LATEST_COLLECTION_TIMESTAMP	DATE	The time the data is collected

### 6.15.3 MGMT\$HA\_BACKUP

MGMT\$HA\_BACKUP displays details of the latest backup of the databases.

**Table 6–159 MGMT\$HA\_BACKUP**

Column	Datatype	Description
HOST	VARCHAR2(256)	Host the database is on
DATABASE_NAME	VARCHAR2(256)	Database target name
TARGET_TYPE	VARCHAR2(64)	Target type of the database. (cluster_database or oracle_database)
DISPLAY_NAME	VARCHAR2(256)	Display name of the target
TARGET_GUID	RAW(16)	The target GUID
SESSION_KEY	NUMBER	Session identifier
SESSION_RECID	NUMBER	Together, with SESSION_STAMP, used to uniquely identify job output from V\$RMAN_OUTPUT
SESSION_STAMP	NUMBER	Together, with SESSION_RECID, used to uniquely identify job output from V\$RMAN_OUTPUT
COMMAND_ID	VARCHAR2(33)	Either a user-specified SET COMMAND ID or a unique command ID generated by RMAN
STATUS	VARCHAR2(23)	One of the following values: <sup>2</sup> RUNNING WITH WARNINGS <sup>2</sup> RUNNING WITH ERRORS <sup>2</sup> COMPLETED <sup>2</sup> COMPLETED WITH WARNINGS <sup>2</sup> COMPLETED WITH ERRORS <sup>2</sup> FAILED
START_TIME	DATE	Start time of the first BACKUP command in the job
END_TIME	DATE	End time of the last BACKUP command in the job
TIME_TAKEN_DISPLAY	VARCHAR2(4000)	Time taken, shown in user-displayable format <nn>h:<nn>m:<nn>s

**Table 6–159 (Cont.) MGMT\$HA\_BACKUP**

Column	Datatype	Description
INPUT_TYPE	VARCHAR2(13)	Contains one of the following values. If the user command does not satisfy one of them, then preference is given in order, from top to bottom of the list. <sup>2</sup> DB FULL <sup>2</sup> RECVR AREA <sup>2</sup> DB INCR <sup>2</sup> DATAFILE FULL <sup>2</sup> DATAFILE INCR <sup>2</sup> ARCHIVELOG <sup>2</sup> CONTROLFILE <sup>2</sup> SPFILE
OUTPUT_DEVICE_TYPE	VARCHAR2(17)	Can be DISK, SBT, or *. An * indicates more than one device (in most cases, it will be DISK or SBT).
INPUT_BYTES_DISPLAY	VARCHAR2(4000)	Values in user-displayable form. They will be converted to a format of nM, nG, nT, nP, and so on.
OUTPUT_BYTES_DISPLAY	VARCHAR2(4000)	Values in user-displayable form. They will be converted to a format of nM, nG, nT, nP, and so on.
OUTPUT_BYTES_PER_SEC_DISPLAY	VARCHAR2(4000)	Output write-rate-per-second. These values are in user-displayable form. They will be converted to a format of nM, nG, nT, nP, and so on.

## 6.16 Storage Reporting Views

### 6.16.1 MGMT\$STORAGE\_REPORT\_DATA

MGMT\$STORAGE\_REPORT\_DATA displays the Storage Data metric attributes which are common across all instrumented Storage Entities.

**Table 6–160 MGMT\$STORAGE\_REPORT\_DATA**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Target Name in Enterprise Manager
TARGET_TYPE	VARCHAR2(64)	Target Type in Enterprise Manager
KEY_VALUE	RAW(20)	Unique Key Value for the Storage Entity
GLOBAL_UNIQUE_ID	RAW(20)	A globally unique persistent identifier for a storage entity. All instances of a shared Storage Entity will have the same global_unique_identifier
NAME	VARCHAR2(256)	Name of the Storage Entity
STORAGE_LAYER	VARCHAR2(32)	Storage layer of the Storage Entity. Example: - OS_DISK - VOLUME_MANAGER - LOCAL_FILESYSTEM - NFS

**Table 6–160 (Cont.) MGMT\$STORAGE\_REPORT\_DATA**

Column	Datatype	Description
ENTITY_TYPE	VARCHAR2(64)	Indicates the type of Entity. Value is vendor specific. Example: Plex, Sub Disk, Diskgroup, Volume group, Metadevice etc.
RAWSIZEB	NUMBER	Total space in bytes
SIZEB	NUMBER	Size in bytes
USEDDB	NUMBER	Used size in bytes
FREEB	NUMBER	Free size in bytes

### 6.16.2 MGMT\$STORAGE\_REPORT\_KEYS

MGMT\$STORAGE\_REPORT\_KEYS displays the relationship between instrumented Storage Entities.

**Table 6–161 MGMT\$STORAGE\_REPORT\_KEYS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Target Name in Enterprise Manager
TARGET_TYPE	VARCHAR2(64)	Target Type in Enterprise Manager
KEY_VALUE	RAW(20)	Unique KEY_VALUE for the Storage Entity
PARENT_KEY_VALUE	RAW(20)	Key value for the parent Storage Entity.

### 6.16.3 MGMT\$STORAGE\_REPORT\_PATHS

MGMT\$STORAGE\_REPORT\_PATHS displays the OS paths for all instrumented storage Entities.

**Table 6–162 MGMT\$STORAGE\_REPORT\_PATHS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Target Name in Enterprise Manager
TARGET_TYPE	VARCHAR2(64)	Target Type in Enterprise Manager
KEY_VALUE	RAW(20)	Unique Key Value for the Storage Entity
NAME	VARCHAR2(256)	Name of the Storage Entity
PATH	VARCHAR2(256)	OS path to the Storage Entity
FILE_TYPE	VARCHAR2(256)	Type of file Examples: - _BLOCKSPECIAL - _DIRECTORY - _REGULAR
STORAGE_LAYER	VARCHAR2(32)	Storage layer of the Storage Entity. Examples: - OS_DISK - VOLUME_MANAGER - LOCAL_FILESYSTEM - NFS

**Table 6–162 (Cont.) MGMT\$STORAGE\_REPORT\_PATHS**

Column	Datatype	Description
ENTITY_TYPE	VARCHAR2(64)	Indicates the type of Entity. Value is vendor specific. Examples: Plex, Sub Disk, Diskgroup, Volume group, Metadevice etc.

#### 6.16.4 MGMT\$STORAGE\_REPORT\_ISSUES

MGMT\$STORAGE\_REPORT\_ISSUES displays the consistency issues encountered when analyzing the instrumented storage metrics.

**Table 6–163 MGMT\$STORAGE\_REPORT\_ISSUES**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Target Name in Enterprise Manager
TARGET_TYPE	VARCHAR2(64)	Target Type in Enterprise Manager
TYPE	VARCHAR2(32)	Type of inconsistency. Value can be one of - MAPPING_ISSUE - MAPPING_WARNING
MESSAGE_COUNT	NUMBER	Count of the number of messages

#### 6.16.5 MGMT\$STORAGE\_REPORT\_DISK

MGMT\$STORAGE\_REPORT\_DISK displays Additional Storage Data Metric Attributes for all Physical Disk Device Storage Entities.

**Table 6–164 MGMT\$STORAGE\_REPORT\_DISK**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Target Name in Enterprise Manager
TARGET_TYPE	VARCHAR2(64)	Target Type in Enterprise Manager
ENTITY_TYPE	VARCHAR2(64)	Indicates the type of Disk Device such as Disk or Disk Partition
USED_PATH	VARCHAR2(256)	The OS path to the disk or partition. If the disk or partition is allocated, then this is the path that is in use.
FILE_TYPE	VARCHAR2(256)	Type of file Examples: - _BLOCKSPECIAL - _REGULAR
SIZEB	NUMBER	Size in bytes
USEDB	NUMBER	Used size in bytes
FREEB	NUMBER	Free size in bytes
VENDOR	VARCHAR2(256)	Name of the disk vendor; detected through SCSI enquiry
PRODUCT	VARCHAR2(256)	Product family from the vendor; detected through SCSI enquiry



## 6.16.6 MGMT\$STORAGE\_REPORT\_VOLUME

MGMT\$STORAGE\_REPORT\_VOLUME displays Additional Storage Data Metric attributes for all Volume Manager Storage Entities.

**Table 6–165 MGMT\$STORAGE\_REPORT\_VOLUME**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Target Name in Enterprise Manager
TARGET_TYPE	VARCHAR2(64)	Target Type in Enterprise Manager
VENDOR	VARCHAR2(256)	Vendor name of the Volume or Software Raid Manager
PRODUCT	VARCHAR2(256)	Vendor name of the Volume or Software Raid Manager
TYPE	VARCHAR2(64)	Indicates the type of Volume entity. It can be vendor specific.  In the case of Veritas Volume Manager for e.g. Volume, Plex, VM Disk, Diskgroup, Sub Disk, Metadevice, Metadevice Partition, Array, Raiddevice, Submirror, Diskset, Slice, raid-disk, spare-disk, Hot spare etc.
DISK_GROUP	VARCHAR2(256)	Disk group or Volume group name
NAME	VARCHAR2(256)	The name of the entity in the volume manager namespace
USED_PATH	VARCHAR2(256)	The OS path to the device. If the device is allocated, then this is the path that is in use.
FILE_TYPE	VARCHAR2(256)	Type of file Examples: - _BLOCKSPECIAL - _REGULAR
RAWSIZEB	NUMBER	In bytes.  For a 2-way mirrored Veritas Volume. It is the sum of the size of each plex.
SIZEB	NUMBER	Size in bytes
USEDDB	NUMBER	Used size in bytes
FREEB	NUMBER	Free size in bytes
CONFIGURATION	VARCHAR2(256)	A string describing the configuration of the Volume.

## 6.16.7 MGMT\$STORAGE\_REPORT\_LOCALFS

MGMT\$STORAGE\_REPORT\_LOCALFS displays Additional Storage Data Metric attributes for all Local Filesystem Storage Entities.

**Table 6–166 MGMT\$STORAGE\_REPORT\_LOCALFS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Target Name in Enterprise Manager
TARGET_TYPE	VARCHAR2(64)	Target Type in Enterprise Manager
FILESYSTEM_TYPE	VARCHAR2(256)	The type of filesystem
FILESYSTEM	VARCHAR2(256)	The filesystem path on the operating system
MOUNTPPOINT	VARCHAR2(256)	The mountpoint path on the operating system

**Table 6–166 (Cont.) MGMT\$STORAGE\_REPORT\_LOCALFS**

Column	Datatype	Description
SIZEB	NUMBER	NUMBER
USEDDB	NUMBER	Used size in bytes
FREEDB	NUMBER	Free size in bytes

### 6.16.8 MGMT\$STORAGE\_REPORT\_NFS

MGMT\$STORAGE\_REPORT\_NFS displays Additional Storage Data Metric attributes for all Network Filesystems.

**Table 6–167 MGMT\$STORAGE\_REPORT\_NFS**

Column	Datatype	Description
TARGET_NAME	VARCHAR2(256)	Target Name in Enterprise Manager
TARGET_TYPE	VARCHAR2(64)	Target Type in Enterprise Manager
FILESYSTEM	VARCHAR2(256)	The filesystem name as seen on the operating system. For NFS filesystems the filesystem name should be in the format nfs_server:/filesystem_name
MOUNTPOINT	VARCHAR2(256)	The mountpoint path on the operating system
SIZEB	NUMBER	Size in bytes
USEDDB	NUMBER	Used size in bytes,
FREEDB	NUMBER	Free size in bytes
NFS_SERVER	VARCHAR2(256)	The server name for the NFS Server.
NFS_SERVER_IP_ADDRESS	VARCHAR2(256)	The IP address of the NFS Server
NFS_VENDOR	VARCHAR2(256)	The NFS Server vendor
MOUNT_PRIVILEGE	VARCHAR2(256)	{READ   WRITE} This is the mount privilege of the filesystem.

---

---

## Fetchlets

Enterprise Manager data retrieval is handled through predefined "fetchlets." A fetchlet is a parametrized data access mechanism that takes arguments (For example, a script, a SQL statement, a target instance's properties) as input and returns formatted data. Each fetchlet handles a specific type of data access. The fetchlets supplied with Enterprise Manager provide data retrieval capability for the most common data access methods, such as SQL, SNMP (Simple Network Management Protocol), HTTP, and DMS (Dynamic Monitoring Service). To handle more complex data access requirements, Enterprise Manager also provides an OS command fetchlet that allows developers to implement custom metric collection methods.

The following fetchlets are supplied with Enterprise Manager:

- [OS Command Fetchlets](#)
- [SQL Fetchlet](#)
- [SNMP Fetchlet](#)
- [URL Timing Fetchlet](#)
- [Dynamic Monitoring Service \(DMS\) Fetchlet](#)
- [HTTP Data Fetchlets](#)
- [URLXML Fetchlet](#)
- [WBEM Fetchlet](#)

### 7.1 OS Command Fetchlets

The operating system (OS) command fetchlets allow you to obtain metric data by executing OS commands (either individually or from scripts) that return a standard out (stdout) data stream.

Three OS command fetchlets are available:

- OS Fetchlet (raw)
- OSLines Fetchlet (split into lines)
- OSLineToken Fetchlet (tokenized lines)

#### 7.1.1 OS Fetchlet

The OS Fetchlet executes a given OS command and returns the command's output in a single cell table.

## Input Parameters

**Table 7–1 OS Fetchlet Input Parameters**

Parameter	Type	Description	Use
command	string	Operating system command to be executed.	required
ENVname	string	OSFetchlet parameters starting with "ENV" appear in the execution environment for the command as <i>name</i> environment variables	zero or more
errStartsWith	String	When defined, this property allows you to define a custom prefix for error messages. If this property is not defined, the OSFetchlet defaults to "em_error=" as the message prefix.	Optional
script	string	Specifies the script to be executed if <i>command</i> property only provides an interpreter. For example, <i>command</i> might consist of "perl." <i>script</i> is then used to specify the particular perl script to be run.  Although scripts can be specified directly from the <i>command</i> parameter, using the <i>script</i> parameter adds to stylistic clarity and readability when defining a target type metadata file.	Optional
args	string	A property that is taken as one or more arguments to the command and script properties.  Although args can be specified directly from the command parameter, using the args parameter adds to stylistic clarity and readability when defining a target type metadata file.	Optional
separateErrorStream	boolean	If an error occurs while executing a <i>command</i> , this property instructs the fetchlet whether to return both the stdout and stderr to the user as an error message. When set to TRUE, only stderr output is sent to the user as an error message when there is a <i>command</i> error. When set to FALSE (default value), both the stdout and the stderr are sent to the user as an error message upon <i>command</i> failure.	Optional. (TRUE/FALSE)
em_metric_timeout	integer	Metric timeout period (in seconds). After the timeout period has finished, the Management Agent returns a timeout exception and terminates any child processes that may have been created. The Management Agent DOES NOT kill any of the grandchild processes.  Specify "-1" for no timeout period.	Optional

### Example

You want to obtain metric data by executing the UNIX *echo* command.

Executing the command from the shell environment, you would enter:

```
echo Line 1|some more|even more\nLine 2\n\nLine 4|a little more|\n|Line 5\n|Line 6|\n|Line 7|again|\nLine 8|the|end
```

The *echo* command produces the following standard output:

```
Line 1|some more|even more
Line 2

Line 4|a little more|
|Line 5
```

```
|Line 6|
|Line 7|again|
Line 8|the|end
```

Using the OS Fetchlet with the given example command.

The fetchlet returns the following 1 x 1 table:

**Figure 7–1 Table Returned by the OS Fetchlet**

```
Line 1|some more|even more
Line 2
Line 4|a little more|
|Line 5
|Line 6|
|Line 7|again|
Line 8|the|end
```

The raw output of the OS command is returned. Any standard error output is appended to the standard output.

### Error Handling

Any problems encountered launching the *command* (For example, the *command* program no longer exists) results in an `oracle.sysman.emSDK.emd.fetchlet.MetricSourceException` wrapping a `java.io.IOException`. If the command exits with a non-zero exit value, the fetchlet throws an `oracle.sysman.emSDK.emd.fetchlet.MetricSourceException` wrapping an `oracle.sysmand.emd.fetchlets.CommandFailedException`.

### Notes

Commands are NOT executed as if they are being run in a shell. This means that common shell symbols do not work, including piping, output redirection, and backgrounding.

Commands cannot read from standard input.

The fetchlet blocks and waits for the command to finish.

## 7.1.2 OSLines Fetchlet (split into lines)

The OS Lines Fetchlet executes a given OS command and tokenizes the OS command's output. The output is tokenized by lines. The fetchlet returns the tokens in a single column table. The *n*th row in the table represents the *n*th line in the OS command's output.

To get the raw, untokenized output of an OS command, use the OS Fetchlet. To get the output of an OS command tokenized by lines and each line tokenized by a given delimiter, see the OS Line Token Fetchlet.

### Input Parameters

**Table 7–2 OSLines Fetchlet Input Parameters**

Parameter	Type	Description	Use
command	string	Operating system command to be executed.	required

**Table 7–2 (Cont.) OSLines Fetchlet Input Parameters**

Parameter	Type	Description	Use
startsWith	string	Only lines starting with this string are included in the result.	optional; default = "" (all lines are included)
ENVname	string	Parameters starting with "ENV" appear in the execution environment for the command as <i>name</i> environment variables	zero or more of these
errStartsWith	string	When defined, this property allows you to define a custom prefix for error messages. If this property is not defined, the OSFetchlet defaults to "em_error=" as the message prefix.	Optional
script	string	Specifies the script to be executed if <i>command</i> property only provides an interpreter. For example, <i>command</i> might consist of "perl." <i>script</i> is then used to specify the particular perl script to be run.  Although scripts can be specified directly from the <i>command</i> parameter, using the <i>script</i> parameter adds to stylistic clarity and readability when defining a target type metadata file.	Optional
args	string	A property that is taken as one or more arguments to the <i>command</i> and <i>script</i> properties.  Although args can be specified directly from the <i>command</i> parameter, using the <i>args</i> parameter adds to stylistic clarity and readability when defining a target type metadata file.	Optional
separateErrorStream	boolean	If an error occurs while executing a command, this property instructs the fetchlet to whether to return both the stdout and stderr to the user as an error message. When set to TRUE, only stderr output is sent to the user as an error message when there is a command error. When set to FALSE (default value), both the stdout and the stderr are sent to the user as an error message upon command failure.	Optional. (TRUE/FALSE)
em_metric_timeout	integer	Metric timeout period (in seconds). After the timeout period has finished, the Management Agent returns a timeout exception and terminates any child processes that may have been created. The Management Agent DOES NOT kill any of the grandchild processes.  Specify "-1" for no timeout period.	Optional

### Example

Let's take the following Unix command:

```
echo Line 1|some more|even more\nLine 2\n\nLine 4|a little more|\n|Line 5\n|Line 6|\n|Line 7|again|\nLine 8|the|end
```

It produces the following output:

```
Line 1|some more|even more
Line 2
```

```
Line 4|a little more|
|Line 5
|Line 6|
|Line 7|again|
Line 8|the|end
```

Running `OSLinesFetchlet` with the given example command produces the following single column table.

**Figure 7–2 Table Returned by the OS LINES Fetchlet**

Line 1 some more even more
Line 2
Line 4 a little more
Line 5
Line 6
Line 7 again
Line 8 the end

Note that without content, "\n" results in a blank line inserted between Line 2 and Line 4.

---

**Notes:** Commands are NOT executed as if they are being run in a shell. This means that common shell symbols do not work, including piping, output redirection, and backgrounding.

Commands cannot read from standard input.

The fetchlet blocks and waits for the command to finish.

The standard output of the command is captured and the standard error is captured and appended to the standard output.

Lines are tokenized using "\n".

---

### 7.1.3 OSLineToken Fetchlet (tokenized lines)

The OS Line Token Fetchlet executes a given OS command and tokenizes the OS command's output. The output is tokenized first by lines, and then each line is tokenized by a given delimiter set. The fetchlet returns the tokens in a table. The *n*th row in the table represents the *n*th line in the OS command's output. The *n*th column in the table represents the *n*th token in a line as determined by the given delimiter set.

To get the raw, untokenized output of an OS command, see the OS Fetchlet.

#### Input Parameters

**Table 7–3 OSLineToken Fetchlet Input Parameters**

Parameter	Type	Description	Use
command	string	Operating system command to be executed.	required
delimiter	string	Set of characters that act as delimiters to tokenize the lines	optional; default = "" (just breaks output into lines)
startsWith	string	Only lines starting with this string are included in the result	optional; default = "" (all lines are included)

**Table 7–3 (Cont.) OSLineToken Fetchlet Input Parameters**

Parameter	Type	Description	Use
ENVname	string	Parameters starting with "ENV" appear in the execution environment for the command as <i>name</i> environment variables	zero or more of these
errStartsWith	String	When defined, this property allows you to define a custom prefix for error messages. If this property is not defined, the OSFetchlet defaults to "em_error=" as the message prefix.	Optional
script	string	Specifies the script to be executed if command property only provides an interpreter. For example, <i>command</i> might consist of "perl." The script is then used to specify the particular perl script to be run.  Although scripts can be specified directly from the command parameter, using the script parameter adds to stylistic clarity and readability when defining a target type metadata file.	Optional
args	string	A property that is taken as one or more arguments to the <i>command</i> and <i>script</i> properties.  Although args can be specified directly from the <i>command</i> parameter, using the args parameter adds to stylistic clarity and readability when defining a target type metadata file.	Optional
separateErrorStream	boolean	If an error occurs while executing a <i>command</i> , this property instructs the fetchlet to whether to return both the stdout and stderr to the user as an error message. When set to TRUE, only stderr output is sent to the user as an error message when there is a <i>command</i> error. When set to FALSE (default value), both the stdout and the stderr are sent to the user as an error message upon <i>command</i> failure.	Optional. (TRUE/FALSE)
em_metric_timeout	integer	Metric timeout period (in seconds). After the timeout period has finished, the Management Agent returns a timeout exception and terminates any child processes that may have been created. The Management Agent DOES NOT kill any of the grandchild processes.  Specify "-1" for no timeout period.	Optional

### Example

Let's take the following Unix command:

```
echo Line 1|some more|even more\nLine 2\n\nLine 4|a little more|\n|Line 5\n|Line 6|\n|Line 7|again|\nLine 8|the|end
```

It produces the following output:

```
Line 1|some more|even more
Line 2
```

```
Line 4|a little more|
|Line 5
|Line 6|
|Line 7|again|
Line 8|the|end
```

Running OSLineTokenFetchlet with the given example command and a single character "|" for the delimiter generates the following table:



**Figure 7-3 Table Returned by the OS Token Lines Fetchlet**

Line 1	some more	even more
Line 2		
Line 4	a little more	
Line 5		
Line 6		
Line 7	again	
Line 8	the	end

### Error Handling

Any problem launching the command (unable to find the command program) results in a `oracle.sysman.emSDK.emd.fetchlet.MetricSourceException` wrapping a `java.io.IOException`.

If the command exits with a non-zero exit value, the fetchlet throws a `oracle.sysman.emSDK.emd.fetchlet.MetricSourceException` wrapping a `oracle.sysman.emd.fetchlets.CommandFailedException`.

### Notes

Commands are NOT executed as if they are being run in a shell. This means that common shell symbols do not work, including piping, output redirection, and backgrounding.

The fetchlet promptly closes the input stream to the running command.

The fetchlet blocks and waits for the command to finish.

Lines are tokenized using "\n".

The delimiter can be a single character or a set of characters. For example, it can be "|+\_ ", if the line should be broken up by pipes, pluses, and underscores. If two or more delimiters are together in the output text, such as "||" or "+|+", then it is as if there are empty string tokens between them. These empty strings get columns in the result table. It is NOT considered that there are empty strings preceding a delimiter that starts a line or following a delimiter that ends a line.

In order to express non-printable characters in the delimiter set (such as tabs) in XML, use "&#xHH;" where H is the hexadecimal identifier for the character.

## 7.2 SQL Fetchlet

The SQL Fetchlet executes a given SQL statement on a given database as a given user and returns the table result.

### Input Parameters

**Table 7-4 SQL Fetchlet Input Parameters**

Parameter	Type	Description	Use
Connection Information			
MachineName	string	Database host	Required
Port	integer	Database port	Required

**Table 7-4 (Cont.) SQL Fetchlet Input Parameters**

Parameter	Type	Description	Use
SID	string	Database SID	Required unless ServiceName is specified
ServiceName	string	Database ServiceName	Required unless SID is specified
OidRepSchemaName	string	Enterprise Manager Management Repository name. This parameter extracts database connection string and credential information (MachineName, Port, SID, ServiceName, UserName, password) from the Management Repository. Using this parameter eliminates the need to specify the connection and credential parameters in each target type metadata file, or having this target instance information exposed in the targets.xml file.	Required if connection and credential information parameters are not used.
OracleHome	string	Database's Oracle Home (used in conjunction with OidRepSchemaName).	Required when OidRepSchemaName is used.
Credential Information			
UserName	string	user name	Required
password	string	user password	Optional; default is ""
Role	string	Role used when connecting to the database (e.g., SYSDBA)	optional; allowed choices are SYSDBA, SYSOPER, and NORMAL (default)
General			
STATEMENT	string	SQL statement or PL/SQL block	Required unless FILENAME is specified.
FILENAME	string	Full path of the file containing the SQL statement or PL/SQL block	Required unless STATEMENT is specified.
NUMROWS	integer	Maximum number of rows to output.	Required
Bind Parameters			
SQLINPARAM<position>	string	Value of input parameter at position <position>	Zero or more, one for each input parameter.
SQLOUTPARAMPOS	integer	Position of output parameter, if it exists	There can be exactly one output parameter, if it exists.
SQLOUTPARAMTYPE	string	Type of the output parameter, if it exists.	There can be exactly one output parameter type, if it exists.
transpose	TRUE/ FALSE	If TRUE, the result is transposed: rows to columns and columns to rows.	

## Notes

The SQL Fetchlet uses the Oracle Call Interface (OCI) from libclntsh.so.

The SQL statement or PL/SQL block can be specified either through the STATEMENT property, or via a file whose name is provided through the FILENAME property.

The SQL fetchlet supports input parameters. Input and output parameters are indicated in the SQL/PLSQL text in the usual way, by using ":<number>". Input parameters can be used to bind values to both SQL queries and PL/SQL blocks.

Input parameter values are specified as properties of the form SQLINPARAM<position>. There can be any number of input parameters. The input parameters need to be scalar: input parameters of type ARRAY and STRUCT are not supported.

The SQL fetchlet supports the execution of anonymous PL/SQL blocks (which may call other functions or procedures) to retrieve data. When executing a block of PL/SQL, data is returned to the fetchlet by means of an OUT parameter. There can be exactly one out parameter. It must be of type SQL\_CURSOR (a PL/SQL REF CURSOR), or it must be a named type that represents an array of objects. In the latter case, each field of the object represents one column of the table; and each object instance in the array represents one complete row in the table. The OUT parameter position and type are specified by means of the properties SQLOUTPARAMPOS and SQLOUTPARAMTYPE. If an OUT parameter is specified, then the fetchlet assumes it is executing PL/SQL and treats the STATEMENT property as an anonymous PL/SQL block.

---

**Note:** When using a SQLOUTPARAMTYPE of type 'ARRAY', you must identify the array as follows:

- If you create the array type specified in the SQLOUTPARAMTYPE from SQL\*Plus or any utility *without* using double quotation marks to surround the array name, then you must specify the array name using all upper-case letters in the target metadata file for this property. The reason for this because the RDBMS automatically changes the array name to all upper-case.
  - If you create the array type specified in the SQLOUTPARAMTYPE from SQL\*Plus or any utility using double-quotation marks to surround the array name, then the RDBMS retains the case specified. For this reason, users must specify the array name using the same case used in the target metadata file.
- 

If no OUT parameter is specified, the fetchlet assumes that it is executing a SQL query.

Note that all input parameters to the SQL fetchlet are strings. This means that all other datatypes will have to be converted to strings. This is straightforward for datatypes such as numbers, but not, for example, dates and timestamps. You can pass an absolute date or timestamp by passing a character representation of the value (using a DateFormat class). There is no way currently to pass in a date function, such as SYSDATE or SYSDATE+1. In such case, you could embed the date argument directly in the SQL, for example:

```
begin func1 (:1, :2, SYSDATE); end;
```

The other caveat is passing null arguments to a procedure. Consider the following SQL:

```
STATEMENT=begin func1(:1,:2); end;
SQLINPARAM1=null
SQLOUTPARAMPOS=2
SQLOUTPARAMTYPE=fooret
```

Assume that the first argument is intended to be a varchar2. By parameterizing it and passing 'null' as the first argument, what we are really doing is passing the \*string\* 'null' to the argument, and not a null value. If you intend to pass a null value, do the following:

```
STATEMENT=begin func1(null, :1); end;
SQLOUTPARAMPOS=1
SQLOUTPARAMTYPE=fooret
```

### Examples

The following properties execute a query (get all users) with no parameters:

```
MachineName=skini-pc
Port=1521
SID=o817
UserName=scott
password=tiger
STATEMENT=select * from all_users;
NUMROWS=30
```

The following properties execute a query (get the first few objects of a specified type owned by a specified user) with input parameters:

```
MachineName=skini-pc
Port=1521
SID=o817
UserName=scott
password=tiger
STATEMENT=select * from all_objects where owner=:1 and object_type=:2 and
rownum<:3tt>
NUMROWS=30
SQLINPARAM1=SYSTEM
SQLINPARAM2=INDEX
SQLINPARAM3=10
```

The following example executes a PL/SQL procedure that returns a cursor and has input parameters:

```
MachineName=skini-pc
Port=1521
SID=o817
UserName=scott
password=tiger
STATEMENT=begin :1 := skini_junk.func1(:2); end;
NUMROWS=30
SQLINPARAM2=SYSTEM
SQLOUTPARAMPOS=1
SQLOUTPARAMTYPE=sql_cursor
```

The following example specifies a PL/SQL procedure that returns an array of strings:

```
MachineName=skini-pc
Port=1521
```

```

SID=o817
UserName=scott
password=tiger
STATEMENT=begin skini_junk.newproc(:1,:2); end;
NUMROWS=30
SQLINPARAM1=SYSTEM
SQLOUTPARAMPOS=2
SQLOUTPARAMTYPE=my_string_array

```

The following example specifies a PL/SQL package that returns an array of structures:

```

MachineName=skini-pc
Port=1521
SID=o817
UserName=scott
password=tiger
STATEMENT=begin :1 := skini_junk.func2(:2,:3,:4,:5,:6); end;
NUMROWS=30
SQLINPARAM2=somename
SQLINPARAM3=someplace
SQLINPARAM4=someanimal
SQLINPARAM5=something
SQLINPARAM6=22
SQLOUTPARAMPOS=1
SQLOUTPARAMTYPE=my_struct_array

```

The PL/SQL used in the examples above is provided below for reference:

```

create or replace type my_type as Object (
    name varchar2(128),
    place varchar2(128),
    animal integer,
    thing number,
    thing2 number);
/
create or replace type my_struct_array as table of my_type;
/

create or replace type my_string_array as table of varchar2(3000);
/

create or replace type my_int_array as table of integer;
/

create or replace package skini_junk as

type Jcr is ref cursor;

function func1(username in varchar2) return Jcr;
function func2(name varchar2, place varchar2, animal integer,
    thing number, thing2 number) return my_struct_array;
procedure newproc(name varchar2, outArray OUT my_string_array);
procedure newproc2(numrows in varchar2, outArray OUT my_int_array);

end skini_junk;
/

create or replace package body skini_junk as

function func1(username in varchar2) return Jcr is

```

```
cr Jcr;
begin
    open cr for select object_name, object_type, status from all_objects where
        owner=upper(username);

    return cr;
end;

function func2(name varchar2, place varchar2, animal integer,
    thing number, thing2 number) return my_struct_array IS
ret my_struct_array := my_struct_array();

begin
    ret.extend(50);

    for i in 1..50 loop
        ret(i) := my_type(name || i,
            place || i,
            animal+i,
            thing+i,
            thing2+i);
    end loop;
    return ret;
end;

procedure newproc(name varchar2, outArray OUT my_string_array) IS
begin
    outArray := my_string_array();
    outArray.extend(100);

    for i in 1..100 loop
        outArray(i) := name || i;
    end loop;
end;

procedure newproc2(numrows in varchar2, outArray OUT my_int_array) IS
begin
    outArray := my_int_array();
    outArray.extend(numrows);
    for i in 1..numrows loop
        outArray(i) := i;
    end loop;
end;

end skini_junk;
/
```

## 7.3 SNMP Fetchlet

An *object identifier* (OID) corresponds to either a MIB variable instance or a MIB variable with multiple instances. Given a list of (OIDs), the SNMP Fetchlet polls an *SNMP agent* on a given host for corresponding instances.

## Input Parameters

**Table 7–5** *SNMP Fetchlet Input Parameters*

Parameter	Type	Description	Use
HOSTNAME	string	Host name of the SNMP agent	Required examples: "bigip.us.oracle.com" "148.87.10.5"
PORT	string	Port of the SNMP agent	Optional. Default is "161"
COMMUNITY	string	SNMP community string	Optional; Default is "public"
TIMEOUT	STRING	SNMP timeout.	Optional. Default is five seconds.
OIDS	string	A list of substrings separated by delimiters. Each substring starts with an OID (in numerical dot notation), and can be optionally ended with *PlacementOID. (See notes for details.)	Required; examples: "1.3.6.1.2.1.2.1.1.1.0,1.3.6.1.2.1.2.1.1.3.0,1.3.6.1.2.1.2.1.1.5.0" "1.3.6.1.2.1.2.1.2.2.1.2 1.3.6.1.2.1.2.1.2.2.1.10 1.3.6.1.2.1.2.1.2.2.1.16" "1.3.6.1.2.1.2.2.1.3 1.3.6.1.2.1.2.2.1.5 1.3.6.1.2.1.4.20.1.1*1.3.6.1.2.1.4.20.1.2 1.3.6.1.2.1.4.20.1.3*1.3.6.1.2.1.4.20.1.2"
DELIM	string	A delimiter to separate individual substrings in OIDS.	Optional; default is whitespace characters (dot), *(star) and 0-9 (digits) cannot be used as delimiters
TABLE	string	Each OID in OIDS corresponds to a variable with multiple instances if this parameter is "TRUE" and to a single variable instance if it is "FALSE".	Optional; default is "FALSE"

## Error Handling

MissingParameterException is thrown if either hostname or OIDS is not given. FetchletException is thrown if TABLE is not equal to either TRUE or FALSE, an I/O error occurs while sending/receiving SNMP messages to/from the agent, or the agent responds with an SNMP error.

## Notes

The table returned by the fetchlet will contain a column for every OID in OIDS. If input OIDs correspond to single variable instances, the table will have just one row with those instances. On the other hand, if the OIDs correspond to variables with multiple instances, each column in the table will contain instances for its OID and each row will correspond to a different *subidentifier*. (A subidentifier is an OID extension that uniquely identifies a particular variable instance for some MIB variable.) OIDS must contain either all OIDs with subidentifiers or all OIDs without the subidentifiers.

For example, to request the variable instances for the three OIDs: sysDescr, sysUpTime, and sysName, OIDS would have to be "1.3.6.1.2.1.2.1.1.1.0 1.3.6.1.2.1.2.1.1.3.0 1.3.6.1.2.1.2.1.1.5.0". In this case, all OIDs contain the instance subidentifier, ".0". The return table would appear as follows (the actual values may be different):

**Figure 7–4** *SNMP Fetchlet*

Sun SNMP Agent, Ultra-4	32504340	necd-view3
-------------------------	----------	------------

Alternatively, assume that some MIB contains the following 3 columns and 4 instances:

**Figure 7-5 SNMP Fetchlet: Columns 3 and 4 Content**

ifDescr (network interface description)	ifInOctets (bytes into an interface)	ifOutOctets (bytes out of an interface)
OID: 1.3.6.1.2.1.2.1.2.2.1.2	OID: 1.3.6.1.2.1.2.1.2.2.1.10	OID: 1.3.6.1.2.1.2.1.2.2.1.16
subidentifier : variable instance	subidentifier : variable instance	subidentifier : variable instance
1: wx1	1: 26150844	1: 29368527
2: wx2	2: 2763185941	2: 3023812977
3: wx3	3: 123615396	3: 2055140730
4: wx4	4: 2068257723	4: 3212899913

To construct a table with 3 columns corresponding to ifDescr, ifInOctets, and ifOutOctets, OIDS would have to look as follows

"1.3.6.1.2.1.2.1.2.2.1.2 1.3.6.1.2.1.2.1.2.2.1.10 1.3.6.1.2.1.2.1.2.2.1.16"

The fetchlet would return:

**Figure 7-6 SNMP Fetchlet:ifDescr, ifInOctets, and ifOutOctets OIDS**

wx1	26150844	29368527
wx2	2763185941	3023812977
wx3	123615396	2055140730
wx4	2068257723	3212899913

The rows correspond to subidentifiers 1,2,3,4 respectively.

Any OID in OIDS can be appended with another *placement* OID. The variable instances for the placement OID do not appear in the returned table. Instead, they determine the place for the variable instances of the original OID within a column. In particular, for every variable instance I with subidentifier S in the set of instances for the original OID, (a) there must exist a variable instance X with subidentifier S in the set of instances corresponding to the placement OID, and (b) X is used as the subidentifier for the instance I.

For example, consider a MIB containing the following 3 columns, each with 4 variable instances:

**Figure 7-7 SNMP Fetchlet: MIB Content with 4 Variable Instances**

ifDescr (network interface description)	ipAdEntNetMask (netmask)	ipAdEntIfIndex (network interface index)
OID: 1.3.6.1.2.1.2.1.2.2.1.2	OID: 1.3.6.1.2.1.2.1.4.20.1.3	OID: 1.3.6.1.2.1.2.1.4.20.1.2
subidentifier : variable instance	subidentifier : variable instance	subidentifier : variable instance
1: wx1	IP1: 255.255.255.0	IP1: 3
2: wx2	IP2: 255.255.128.0	IP2: 1
3: wx3	IP3: 255.255.255.240	IP3: 4
4: wx4	IP4: 255.255.254.0	IP4: 2

To construct a table containing ifDescr and ipAdEntNetMask, OID of ipAdEntIfIndex would have to be used as the placement OID to "align" the columns. Thus, the OIDS



input to the fetchlet would be "1.3.6.1.2.1.2.2.1.2 1.3.6.1.2.1.2.1.4.20.1.3\*1.3.6.1.2.1.2.1.4.20.1.2". The fetchlet output will be as follows:

**Figure 7–8 SNMP Fetchlet: Table Containing ifDescr and ipAdEntNetMask**

wx1	255.255.128.0
wx2	255.255.254.0
wx3	255.255.255.0
wx 4	255.255.255.240

If OIDS were "1.3.6.1.2.1.2.2.1.2 1.3.6.1.2.1.2.1.4.20.1.3" for the previous example, the output would be as follows:

**Figure 7–9 SNMP Fetchlet: Alternate OID**

wx1	
wx2	
wx3	
wx 4	
	255.255.255.0
	255.255.128.0
	255.255.255.240
	255.255.254.0

## 7.4 URL Timing Fetchlet

The URL Timing Fetchlet gets the contents of a given URL, timing not only the base page source but any frames or images in the page as well.

### Input Parameters

**Table 7–6 URL Timing Fetchlet Input Parameters**

Parameter	Description	Use
url#	URL(s) to download. "url0" is required but any number of URLs can be specified beyond url0 that following the convention: url0, url1, url2, url3.	Required.
proxy_host	Proxy host used to make a URL connection.	Optional. Specifies the proxy to be used for accessing URLs. If the proxy_host_override value is provided, then that value will be used instead.
proxy_port	Port used by the proxy host used make the URL connection.	Optional.
dont_proxy_for	Domains for which the proxy will not be used.	Optional. For example, .us.oracle.com, .uk.oracle.com

**Table 7–6 (Cont.) URL Timing Fetchlet Input Parameters**

<b>Parameter</b>	<b>Description</b>	<b>Use</b>
use_proxy	When used in conjunction with the proxy override input parameters, use_proxy specifies a proxy to be used in lieu of the original proxy. When set to false without the proxy override parameters set, no proxy is used.	Optional. Parameter can be set to true or false.
proxy_host_override	Alternate proxy host used to make the URL connection.	Optional. Overrides proxy_host.
proxy_port_override	Alternate proxy port used to make the URL connection.	Optional. Overrides proxy_port.
dont_proxy_override	Do not use the proxy for domains.	Optional. Parameter can be set to true or false.
internet_cert_loc	Path pointing to the location of a certificate to be used to access a secure (HTTPS) URL.	Optional.
auth_realm	Realm for the Basic Authentication log on. If the realm is not specified for the authentication, authentication does not occur and the download of the page fails with a 401 response code.	Optional.
auth_user	Username for Basic Authentication.	Optional.
auth_password	Password for Basic Authentication.	Optional.
retries	Number of times to retry the url if it initially fails.	Optional. Default = 1
connection_timeout	Wait time (in milliseconds) allowed to establish a connection to a server. This time also includes time required for name resolution.	Optional. Default= 60000 milliseconds (1 minute)
read_timeout	Idle time in the read waiting for the server to respond. For example, if no data is received from the server during the specified timeout period, the operation is considered failure.	Optional. Default = 12000 milliseconds (2 minutes)
timeout	Number of milliseconds after which the page download is considered a failure. This will detect if the site is functional but is extremely slow.	Optional. Default = 300000 ms (5 minutes)
status_comparator	When collating the rows to make a single row, the status_comparator parameter will indicate whether all URLs should have been a success (and) or any URLs should have been a success (or).	Optional. Default = and
cache	Indicates whether to use a cache when accessing an URL. Set the parameter to "n" to specify that no cache be used.	Optional. Default = y Note: The scope of the cache is per request. There is no persistent cache across multiple get metric requests.

**Table 7-6 (Cont.) URL Timing Fetchlet Input Parameters**

Parameter	Description	Use
output_format	Specifies the output format to be used: summary, detailed, repeat_column. For more information on output formats, see <a href="#">Metric Columns and Output Modes</a> on page 7-17.	Required. summary : gives a default set of metrics in a single row for all urls  detailed: gives a default set of metrics for each url.  repeat_column : gives a single row of metric with timing for each of the url.
metrics	Specifies which metric columns need to be returned. For more information on metrics columns returned for each output format, see <a href="#">Table 7-8, "URLTIMING Fetchlet: Metric Columns"</a>	Optional. Allows you to specify of what needs to be returned from the fetchlet and in which order.  Example: status, status_description, total_response_time

### Metric Columns and Output Modes

The format of information and specific metric information returned are controlled by the "output\_format" and "metrics" input parameters. The following table lists the format categories and the metrics (columns) returned by each. For a description of available metric columns, see [Table 7-8, "URLTIMING Fetchlet: Metric Columns"](#)

**Table 7-7 URLTIMING Fetchlet: Output Formats**

Output Format	Description	Metric Columns
summary	Returns a default set of metrics in a single row for all URLs  If the metrics input parameter is specified, then only the columns specified will be returned.	computed_response_time, status, status_description, dns_time, connect_time, redirect_time, first_byte_time, html_time, content_time, total_response_time, rate, max_response_time, avg_response_time, avg_connect_time, avg_first_byte_time, broken_count, broken_content
detailed	Returns a default set of metrics for each url.  If the metrics input parameter is specified, then only the columns specified will be returned.	url, computed_response_time, status, status_description, dns_time, connect_time, redirect_time, first_byte_time, html_time, content_time, total_response_time, rate, redirect_count, html_bytes, content_bytes, total_bytes, avg_connect_time, avg_first_byte_time, broken_count, broken_details

**Table 7-7 (Cont.) URLTIMING Fetchlet: Output Formats**

Output Format	Description	Metric Columns
repeat_column	Returns a single row of metrics with timing for each of the URLs.  If the metrics input parameter is specified, then those columns will be returned for each url followed by overall status and status_description. (Note the output will always be single row).	total_response_time repeated for each URL followed by overall status and status_description.

**Metric Columns**

The following table shows the metric columns returned by the URLTIMING fetchlet.

**Table 7-8 URLTIMING Fetchlet: Metric Columns**

Column Name	Description
status	The overall status of all URLs. By default AND is used to compute overall status but this can be changed using the status_comparator input parameter.
connect_time	The time to connect to the server and send the request.
first_byte_time	Time taken between sending the request and reading the first byte from the response.
total_response_time	Time taken for fetching ALL urls and associated content (gif, css, javascript etc).
max_response_time	Also referred as Slowest page time. This the time taken by the slowest URL.
avg_response_time	Average response time for URL. Computed as total response time / number of pages (urls).
rate	Kilo Bytes per second. This is computed by total bytes received / total time taken to receive them.
html_time	Total time taken to download the html part of all pages. This time excludes time to fetch images and other contents. (Includes time to fetch frame html).
content_time	Time taken to download the page content (gif, javascripts, css etc.).
redirect_time	Total time taken for all redirects occurring while fetching the set of urls specified.
redirect_count	Number of redirects.
total_bytes	Total number of bytes.
html_bytes	Total number of HTML bytes. (Includes bytes for frame html).
content_bytes	Total number of content bytes.
status_description	This is present only when the status is down. Corresponds to HTTP Status description.
request_count	Number of request made. (Includes all html as well as content requests).
broken_count	Number errors while fetching images or other content elements.

**Table 7–8 (Cont.) URLTIMING Fetchlet: Metric Columns**

Column Name	Description
broken_details	List of images or other content elements that could not be fetched. This has format of url[broken list], url[broken list...
computed_response_time	This time approximates the time it would have taken for a client (like browser) to fetch all the pages in the transaction. This number is computed as if the contents of every page (gifs, css etc) were fetched using multiple threads.
avg_connect_time	Total connect_time / total number of connections made.
avg_first_byte_time	Total First Byte Time / Number of requests made (either to fetch HTML or content).
dns_time	Time to resolve host name (not implemented, always returns zero).
url	Returns the url, can only be used in 'detailed' output_format.

**Example**

Let's take the following URL:

```
url0=http://www.oracle.com/
```

With the input parameter `output_format=summary`, the fetchlet returns the following table (minus the headers on the columns):

**Figure 7–10 Summary Output Format**

computed response time	status	status description	dns time	connect time	redirect time	first byte time	html time	content time	total response time	rate (Kbytes per second)	max response time	avg response time	avg connect time	avg first byte time	broken count	broken content
540	1		0	548	0	149.0	1.0	7.0	705	95.16	705	705	54.80	14.90	0	

With `output_format = summary` and `metrics = total_response_time, status, status_description` the fetchlet returns the following table (minus the headers on the columns):

**Figure 7–11 Summary Output Format with Specified Metric Columns**

total response time	Status	Status description
705.0	1	

With `output_format = summary` and `metrics = total_response_time, status, status_description` the fetchlet returns the following table (minus the headers on the columns) and the server is giving error:

**Figure 7–12 Summary Output Format with Specified Metric Columns and Internal Server Error**

total response time	status	status description
	0	Internal Server Error -- http://www.oracle.com

Let's take the following URL:

url0=http://www.oracle.com/  
url1=http://nedc.us.oracle.com/

With the output\_format=summary, the fetchlet returns the following table (minus the headers on the columns). Here the numbers are time taken for fetching both the urls.

**Figure 7–13 Summary Output Format for Two URLs**

computed response time	status	status description	dns time	connect time	redirect time	first byte time	html time	content time	total response time	rate (Kbytes per second)	max response time	avg response time	avg connect time	avg first byte time	broken count	broken content
4344	1		0	603	0	7277	438	318.0	8636	16.92	8098	4318	22.33	269.52	0	

With the output\_format=detailed, the fetchlet returns the following table (minus the headers on the columns):

**Figure 7–14 Detailed Output for Two URLs**

url	uri	computed response time	status	status description	dns time	connect time	redirect time	first byte time	html time	content time	total response time	rate	redirect count	html bytes	content bytes	total bytes	avg connect time	avg first byte time	broken count	broken content
http://www.oracle.com/	not supported	531.0	1			548	0	149.0	1	7	705	95.16	0	18207	48883	67090	54.80	14.90	0	
http://nedc.us.oracle.com/	not supported	4424.0	1			480	0	6227.0	442	230	7379	10.71	0	24627	54427	79054	28.24	366.29	0	

With the output\_format=repeat\_column, the fetchlet returns the following table (minus the headers on the columns):

**Figure 7–15 Repeat Column Output Format**

total response time(oracle.com)	total response time (nedc)	status	status description
705.0	7379	1	

### Error Handling

Metric error if the URL parameter is missing, malformed, or if the metric cannot be computed.

**Notes**

The time required to perform a retry will be added on to the total time of the page. For example, if two retries are performed and then a success occurs, the total page time will be the time of the page that succeeded plus the time it took for the two retries to fail.

**Proposed usage:**

- For basic monitoring:  
Use `url0=<URL to be monitored>` , `output_mode=summary` and specify `metrics=status, computed_response_time, status_description`
- For getting all columns:  
Use `url0=<url to be monitored>` , `output_mode=summary`

## 7.5 Dynamic Monitoring Service (DMS) Fetchlet

The Dynamic Monitoring Service (DMS) fetchlet contacts an Application Server (AS) and then collects the metrics instrumented by the DMS.

The DMS allows application and system developers to measure and export customized, component-specific performance metrics. The Oracle Management Agent allows software components to import runtime performance data into Oracle Enterprise Manager Grid Control.

The DMS Fetchlet is an Oracle Management Agent plug-in module that allows the Management Agent to import the performance data that is exported by the DMS. Using the DMS fetchlet, any component that is instrumented using DMS API calls may share its performance data with Enterprise Manager Grid Control.

### 7.5.1 Advantages to Using DMS for Oracle Management Agent Integration

With DMS, a component can insulate itself from the operational details of the Management Agent. A component would not need to deploy (or maintain) its own fetchlet or deploy (or maintain) a Tcl script or shell script to plug into one of the existing fetchlets. A component would not need to devise its own new way of measuring or exporting performance metrics. Performance metrics can be measured and reported in a consistent way across components. The DMS fetchlet contacts the remote DMS runtime directly with no need for forking shell scripts or Tcl scripts. Most importantly, DMS automatically produces the long, complicated metadata document for you and thereby saves many hours of tedious and error-prone hand editing.

**Input Parameters****Table 7-9 DMS Fetchlet Input Parameters**

Name	Type	Description	Use
oraclehome	String	Top directory under which the monitored IAS instance is installed. It is used only for monitoring local IAS processes. For monitoring remote IAS processes, users should give it an empty value and specify property "opmnremoteport" and/or "machine" instead.	Required. Example: "/private/oracle/ias"

**Table 7–9 (Cont.) DMS Fetchlet Input Parameters**

<b>Name</b>	<b>Type</b>	<b>Description</b>	<b>Use</b>
version	String	AS Version number of the target. It is used to distinguish the version of monitored AS instance.	Optional Example: "9.0.4"
opmnport	Integer	Oracle Process Monitoring and Notification (OPMN) port. It is used primarily for monitoring remote AS processes. It should be specified together with property "machine". If it is present and valid, property "oraclehome" and "httpport" are ignored.	Optional Example: "6200"
httpport	Integer	HTTP port is used primarily for monitoring stand-alone processes. It should be specified together with property "machine". It will be ignored, if property "opmnport" is present. If it is present and valid, property "oraclehome" is ignored.	Optional Example: "7777"
machine	String	Host name where the Internet Application Server (AS) instance runs. It should be specified together with property "opmnport". If it is not present, the local host is assumed.	Optional Example: "my-sun.us.oracle.com"
metric	String	Name of the table-type metric.	Required Example: "Servlets"
columnOrder	String	A list of metric column names separated by ";". The column names must be specified in same order as they appear in the target type metadata file.  Do not include "name", "host", "process" and "fullname" columns.	Required Example: "processTimes;totalRequest;requestRate"
usecache	String	Whether to cache this metric. The legal values are "true", "false" and "refreshall" with "true" being the default. The "refreshall" value tells the DMS to delete its cache data and retrieve the most recent data from all targets.	Optional. Example: "false" Setting "usecache" to "false" will bypass DMS caching
proxyHost	String	Proxy host through which to make the HTTP connection	Optional Example: "proxy.us.oracle.com"
proxyPort	Integer	Proxy port through which to make the HTTP connection	Optional Example: "80"
dontProxyFor	String	Domains for which the proxy will not be used.	Optional Example: ".us.oracle.com" or "18.219.0"



**Table 7–9 (Cont.) DMS Fetchlet Input Parameters**

Name	Type	Description	Use
useDefaultProxy	String	When used in conjunction with the proxy override parameters, this variable specifies a proxy other than the original one. When set to false without the proxy override parameters set, no proxy at all is used.	Optional Example: "true" or "false"
proxyHostOverride	String	proxy host through which to make the HTTP connection	Optional Example: "www-proxy.us.oracle.com"
proxyPortOverride	Integer	proxy port through which to make the HTTP connection	Optional Example: "80"
authrealm	String	Realm for the Basic Authentication logon. If the realm is not specified for the authentication, authentication does not occur and the download of the page fails with a 401 response code.	Optional Example: "Please input your flex account login:"
authuser	String	Username for Basic Authentication	Optional "superuser"
authpwd	String	Password for Basic Authentication	Optional Example: "welcome"

**Error Handling**

DMS Fetchlet throws `MissingParameterException` if any of the properties "oraclehome", "metric", "columnOrder", "opmnport", or "httpport" is missing. It throws `FetchletException` if any of the ports given is not valid.

**Notes**

The first four columns of the metric table returned are always column "name", "fullname", "host" and "process". Therefore, do not include them in columnOrder string. Property "machine" should be specified together with either properties "opmnport" or "httpport". In this case, the property "oraclehome" is ignored.

**7.5.2 DMS Fetchlet/Oracle Management Agent Integration Instructions**

DMS has been used in several components (such as Apache, JServ, OSE, and Portal) to provide a consistent performance monitoring infrastructure for Oracle 9i Application Server. The Sensors are easy to use and save most of the work related to performance measurement because they hide most of the details related to timing, counting, and categorization. Finally, DMS hides many Management Agent details from component developers and much of the Management Agent integration effort.

### 7.5.2.1 Integrating DMS Data with the Management Agent

As mentioned earlier, DMS allows application and system developers to measure and export customized, component-specific performance metrics. The Oracle Management Agent enables software components to import runtime performance data into Enterprise Manager Grid Control. This section describes how to integrate DMS performance metrics with the Management Agent.

#### Step 1: Install AS

#### Step 2: Install Enterprise Manager Grid Control

#### Step 3: Instrument your Component with DMS

To enable DMS metrics for Enterprise Manager Grid Control, you must follow two additional rules:

- **Rule 1: All Nouns exported to the Management Agent must have types**  
Noun types can be set either by specifying the "type" parameter in the Noun.create() methods or by using the Noun.setType(String) method. The idea is that every Noun type will be converted automatically to a management repository table. Every Noun of a given type will become a row in the type's corresponding management repository table. The metrics contained by a Noun become columns in the repository table metric. Any Noun without a type will not be exported to Management Agent.
- **Rule 2: All Nouns of a given type must contain a consistent set of Sensor names**  
Because the metrics contained by a Noun become columns in a management repository table, you must make sure that all Nouns of a given type contain the same Sensors. This ensures that each row of the corresponding repository table has the same set of columns. DMS does not check this constraint for you.

For example, the following Java snippet shows how to create typed Nouns that contain a consistent set of Sensors. DMS will automatically convert these into a repository table named "MyType":

```

/* first create the nouns*/
Noun n1 = Noun.create("/myExample/myComponent/noun1", "MyType");
Noun n2 = Noun.create("/myExample/myComponent/noun2", "MyType");

/* next, create the Sensors */
PhaseEvent pe1 = PhaseEvent.create(n1, "criticalPhase", "a critical interval");
PhaseEvent pe2 = PhaseEvent.create(n2, "criticalPhase", "a critical interval");
Event e1 = Event.create(n1, "importantEvt", "an important event");
Event e2 = Event.create(n2, "importantEvt", "an important event");

/* here is a third set that shows the use of Noun.setType(String) */
PhaseEvent pe3 = PhaseEvent.create(
    "/myExample/myComponent/noun3/criticalPhase",
    "a critical interval");
Event e3 = PhaseEvent.create(
    "/myExample/myComponent/noun3/importantEvt",
    "an important event");
Noun n3 = Noun.get("/myExample/myComponent/noun3");
n3.setType("MyType");

```

For this example, the "MyType" table will contain three rows and four columns. Besides the columns corresponding to the two Sensors, there will be a "name" column and a "path" column that will contain the DMS path name including the process name and "/myExample/myCom...".

If these Nouns/Sensors are created in several servlet engines within the AS site, then the AggreSpy will find each of the servlet engines and will aggregate all of the Nouns/Sensors into a single MyType table.

#### Step 4: Generate your Target Metadata Document

You can generate the Target Metadata Document using your browser. Point your browser to your AS site that you want to monitor using the following URL:

```
http://YOUR_AS_HOST:YOUR_AS_PORT/YOUR_SERVLET_PATH/AggreSpy?format=targetmetadata
```

You should use the actual host, port and servlet path of your AS installation in the above URL. The servlet path usually defaults to "servlet". The XML document you get is the Target Metadata Document for your AS site. The first comment of the XML document explains where you can obtain the Target Metadata Document and instructions telling you what needs to be done to this document.

#### Step 5: Install the Target Metadata Document

Follow the steps described in the first comment of the XML document. Save the XML document to a file called "oracle\_dms.xml" under the "metadata" directory of your Enterprise Manager installation (OMS\_ORACLE\_HOME/sysman/admin/metadata/). If you want to monitor a subset of the metrics or merge the metrics with the ones in the existing "oracle\_dms.xml" file, you should save this new definition to a separate file called target\_name.xml. You will also need to change the Target Type entry in the generated metadata document.

Next, you should add the target instance information of your AS site to your "targets.xml" file residing under the top directory of your Enterprise Manager installation. You can find a block of XML tags in the comment you read. They look like:

```
<Target Type='oracle_dms' NAME='DMS_YOUR-IAS-HOST_YOUR-IAS-PORT' VERSION='2.0'>
  <Property NAME='host' VALUE='YOUR_IAS_HOST' />
  <Property NAME='port' VALUE='YOUR_IAS_PORT' />
  <Property NAME='dmsPath' VALUE='YOUR_SERVLET_PATH' />
</Target>
```

Copy this block and paste it to the "targets.xml" file between <targets> and </targets> tags.

Finally, to add the new target metadata file and target instance information from the targets.xml file to Enterprise Manager Grid Control, you must run the following command:

```
>$ORACLE_HOME/bin/emctl reload
```

#### Step 6: View your metrics

You are ready to view your metrics using Enterprise Manager's Metric Browser. See ["Activating the Metric Browser"](#) on page 2-10 for information on setting up the Metric Browser. First, make sure that AS and your component are still running. Next, restart the Oracle Management Agent. Finally, point your browser to your Management Agent installation using the following URL:

```
http://<YOUR_AGENT_HOST>:<YOUR_AGENT_PORT>/emd/browser/main
```

The Management Agent port information can be found in the \$AGENT\_HOME/sysman/config/emd.properties file at the EMD\_URL line.

You should use the actual host and port of your Management Agent installation in the above URL. You will find your AS site listed as the target "DMS\_YOUR-AS-HOST\_YOUR-AS-PORT". If you click on the link, you will see a list of metric IDs. You can browse your metrics by clicking on the respective metric IDs.

## 7.6 HTTP Data Fetchlets

The HTTP data fetchlets obtain the contents of a URL and returns the contents of the URL as data. Three fetchlets are available:

- URL Fetchlet
- URL Lines
- URL Lines Token

### 7.6.1 URL Fetchlet (raw)

The URL Fetchlet gets the contents of a given URL and returns the contents of the URL in a single cell table.

To get the output of a URL tokenized by lines and each line tokenized by a given delimiter, see the URL Line Token Fetchlet.

#### Input Parameters

**Table 7–10 URL Fetchlet Input Parameters**

Name	Description	Use
url	URL to retrieve the contents of	required
proxyHost	proxy host through which to make the URL connection.	optional
proxyPort	proxy port through which to make the URL connection.	optional

#### Example

Let's take the following URL:

`http://localhost/nhcities.txt`

It has the following contents:

*Line 1:* Nashua, Keene,

*Line 2:* Concord

*Line 3:* , Conway, Manchester, Milford, Brookline,

*Line 4:*

*Line 5:* Hollis, Meredith

Now let's run the URL Fetchlet with the given URL. The fetchlet returns the following one-by-one table:

**Figure 7–16 URL Fetchlet Output**

```
Nashua, Keene,
Concord
, Conway, Manchester, Milford, Brookline,
Hollis, Meredith
```

The raw contents of the URL is returned.

### Error Handling

MissingParameterException if URL parameter is missing.

FetchletException if the URL is malformed or an I/O error occurs in retrieving the content of the URL.

## 7.6.2 URL Lines Fetchlet (split into lines)

The URL Fetchlet gets the contents of a given URL and tokenizes the contents of the URL. The output is tokenized by lines. The fetchlet returns the tokens in a single column table. The nth row in the table represents the nth line of the URL contents.

---

**Note:** To get the raw, untokenized contents of a URL, see the URL Fetchlet. To get the contents of a URL tokenized by lines and each line tokenized by a given delimiter, see the URL Line Token Fetchlet.

---

**Table 7–11 URL Lines Fetchlet Input Parameters**

Name	Description	Use
url	URL to retrieve the contents of	required
proxyHost	proxy host through which to make the URL connection.	optional
proxyPort	proxy port through which to make the URL connection.	optional
startsWith	only lines starting with this string are included in the result	optional; default = "" (all lines are included)

### Example

Let's take the following URL:

```
http://localhost/nhcities.txt
```

It has the following contents:

*Line 1:* Nashua, Keene,

*Line 2:* Concord

*Line 3:* , Conway, Manchester, Milford, Brookline,

*Line 4:*

*Line 5:* Hollis, Meredith

Now let's run the URL Fetchlet with the given URL.  
 The fetchlet returns the following table:

**Figure 7–17 URL Lines Fetchlet Output**

Nashua, Keene,
Concord
, Conway, Manchester, Milford, Brookline,
Hollis, Meredith

**Error Handling**

MissingParameterException if URL parameter is missing.  
 FetchletException if the URL is malformed or an I/O error occurs in retrieving the content of the URL.

**Notes**

Lines are tokenized using "\n".

**7.6.3 URL Line Token Fetchlet (tokenized lines)**

The URL Fetchlet gets the contents of a given URL and tokenizes the contents of the URL. The output is tokenized first by lines, and then each line is tokenized by a given delimiter set. The fetchlet returns the tokens in a table. The *n*th row in the table represents the *n*th line of the URL content. The *n*th column in the table represents the *n*th token in a line as determined by the given delimiter set.

To get the raw, untokenized contents of a URL, see the URL Fetchlet.

**Table 7–12 URL Line Token Fetchlet Input Parameters**

Name	Description	Use
url	URL to retrieve the contents of	required
delimiter	set of characters that act as delimiters to tokenize the lines	optional; default = "" (just breaks output into lines)
proxyHost	proxy host through which to make the URL connection.	optional
proxyPort	proxy port through which to make the URL connection.	optional
startsWith	only lines starting with this string are included in the result	optional; default = "" (all lines are included)

**Example**

Let's take the following URL:  
`http://localhost/nhcities.txt`  
 It has the following contents:  
 Line 1: Nashua, Keene,

Line 2: Concord

Line 3: , Conway, Manchester, Milford, Brookline,

Line 4:

Line 5: Hollis, Meredith

Now let's run the URL Fetchlet with the given URL and a single character "," for the delimiter.

The fetchlet returns the following table:

**Figure 7-18 URL Token Lines Output**

Nashua	Keene		
Concord			
Conway	Manchester	Milford	Brookline
Hollis	Meredith		

### Error Handling

MissingParameterException if URL parameter is missing.

FetchletException if the URL is malformed or an I/O error occurs in retrieving the content of the URL.

### Notes

Lines are tokenized using "\n".

The delimiter can be a single character or a set of characters. For example, it can be "|+\_ ", if the line should be broken up by pipes, pluses, and underscores. If two or more delimiters are together in the output text, such as "||" or "+|+", then it is as if there are empty string tokens between them. These empty strings get columns in the result table. It is NOT considered that there are empty strings preceding a delimiter that starts a line or following a delimiter that ends a line.

In order to express non-printable characters in the delimiter set (such as tabs) in XML, use "&#xHH;" where H is the hexadecimal identifier for the character.

## 7.7 URLXML Fetchlet

The URL XML Fetchlet obtains the XML content of a given URL, and extracts information based on a given pattern. A pattern is a list of "chunks" of XML to match against. The return table is a table with a column for each grabber (\*) in the pattern in order and a row each time the pattern chunk matches in the XML content.

### Input Parameters

**Table 7-13 URLXML Fetchlet Input Parameters**

Name	Description	Use
url	URL to retrieve the contents of	Required.

**Table 7–13 (Cont.) URLXML Fetchlet Input Parameters**

Name	Description	Use
pattern	The pattern used to extract information from XML; this is a list of XML chunks that is compared against the XML content of the URL. Each chunk contains one or more "grabbers" (*) in the text portion of the elements that define what should be flattened into text and extracted.	Required.
proxyHost	The proxy host through which to make the URL connection.	Optional.
proxyPort	The proxy port through which to make the URL connection.	Optional.
ignoreDtd	If set to TRUE, specifies that the DTD file referenced by the content XML should be ignored. This is useful in cases where the DTD file cannot be accessed.	Optional.
generateKey	If set to true, a unique key will be generated for each row. The key will occupy the first column of the result, and will be numeric.	Optional.
throwConnException	If set to TRUE, a java.net.ConnectException will be thrown. Otherwise, it will be caught and an empty result set will be returned. Setting this property to FALSE provides behavior which is consistent with the DMSFetchlet.	Optional. The default value is TRUE.

### Example

Let's take the following URL:

```
http://localhost/urlxmltestfile.xml
```

It has the following content:

```
<?xml version="1.0"?>
<testfile>
  <test>Simple text</test>
  <test><level>A little more complex</level></test>
  <test></test>
  <notatest></notatest>
  <test>Yet more complexity<level>Even a little more complex</level>Will it ever
stop?</test>
  <test1>must match<level>extract me!</level></test1>
  <test1>must match here<level>extract me, too!</level></test1>
</testfile>
```

Running the URL XML Fetchlet with the given URL and the pattern:

```
<testfile><test>*<level>*</level></test></testfile>
```

returns the following table:

**Figure 7–19 URL XML Fetchlet Output**

A little more complex	A little more complex
Yet more complexityEven a little more complexWill it ever stop?	Even a little more complex



## Error Handling

MissingParameterException if URL or pattern parameters are missing.

A FetchletException is generated if:

- The URL is malformed.
- An I/O error occurs in retrieving the content of the URL.
- The URL contents or pattern contains invalid XML.

## Notes

Setting the proxy host and/or port changes these settings for the java.net package for the whole Java environment and is not thread-safe if the proxy settings are changing.

## 7.8 WBEM Fetchlet

The WBEM fetchlet accesses a CIMOM and retrieves requested information using the specified CIM class. The CIM class is mapped to a Management Repository table metric. The name of the CIM class is the name of the table metric that is returned, and the properties defined for the CIM class are used to name the table columns for the metric. The properties of interest must be specified during metric definition.

The fetchlet returns the instances that have been instantiated for the CIM class as rows of the Management Repository table metric.

### Input Parameters

**Table 7-14** WBEM Fetchlet Input Parameters

Name	Type	Description	Use
hostname	String	Host name of the CIMOM	Optional; default is "localhost"
port	Integer	Port for the CIMOM	Optional; default is 5988
namespace	String	CIM Namespace	Optional; default is "root/cimv2"
username	String	Username to use for CIMOM authorization on the host where the CIMOM is running	Required
password	String	Password to use for CIMOM authorization on the host where the CIMOM is running	Required
CIMclassname	String	Name of the CIM class whose instances will be returned	Required for all operations except STATUS. STATUS operations just check whether the CIMOM is running, so a class name is not needed.

**Table 7–14 (Cont.) WBEM Fetchlet Input Parameters**

<b>Name</b>	<b>Type</b>	<b>Description</b>	<b>Use</b>
operation	String	Operation to be performed. Supported operations include COUNT, which returns a count of the number of instances in the class, VALUES, which returns the values of the specified properties for each instance of the class, or STATUS, which provides status information about the CIMOM.	Optional, default is VALUES
properties	String	The property names from the CIM class definition that we are interested in collecting.	Required for VALUES operation. If the operation is VALUES, we can have 1 to N of these, separated by a semicolon. If the operation is VALUES, and no properties are provided, an error is returned. Properties are handed to the EMD in the order that they are specified.

**Error Handling**

The following types of errors have been identified for the WBEM fetchlet.

**MissingParameterException occurs when:**

- No CIM Class parameters match.

**Fetchlet exception occurs when:**

- The class name is not found in the CIMOM namespace.
- The namespace is not found.
- The connection to the CIMOM does not have valid credentials.
- The connection to the CIMOM failed because the CIMOM was not running.
- The CIM class property does not exist
- An unsupported operation was specified
- No properties were specified.

**Notes**

**Ports:** Some CIMOM client interfaces expose the port that the CIMOM is listening on while some clients do not. To cover both cases, the port is exposed as an optional input parameter that defaults to port 5988. This is the default Pegasus CIMOM listener port. The Java API that is provided through Sun's Wbem Services does not expose the CIMOM port.

**Protocols:** Most CIMOMs support either an RMI or HTTP protocol for communicating with the CIMOM. The testing that has been done shows that the HTTP protocol is not as stable, and in some cases, not fully implemented in the CIMOM. Because of this, the protocol currently defaults to RMI. The actual parameters for the WBEM Services CIMOM for the protocol are: CIMClient.CIM\_RMI or CIMClient.CIM\_XML.

Fetchlet Operations: The WBEM APIs are very flexible at allowing clients to traverse the class hierarchies that are defined and their associations. At this point in time, the options on accessing CIM data from an EMD are restricted to counting, getting the properties of classes, and CIMOM status. These are the more important operations that need to be performed for monitoring. As additional requirements come in, we can add new operations to support them if necessary. For the prototype, only the count operation has been implemented.

Authentication: Most CIMOMs provide APIs to support authentication through a user identity mechanism. The majority of the CIMOMs have not implemented the API, so this capability is really a no-op. In any case, we've supplied the capability in the fetchlet so that as CIMOM implementations catch up with the standard, we'll have the necessary support in place.

### Examples

The Wbem fetchlet supports three basic operations. At this point, the fetchlet only handles one operation at a time, so you cannot mix count, status, and value operations within a single fetchlet call. The following example shows how to write the metadata for a COUNT operation:

#### Example 7-1 COUNT Operation Metadata

```
<Metric NAME="Load" TYPE="TABLE">
  <Display>
    <Label NLSID="wbem_cimom_load">Load</Label>
  </Display>
  <TableDescriptor>
    <ColumnDescriptor NAME="Active Clients" TYPE="NUMBER" IS_KEY="FALSE">
      <Display>
        <Label NLSID="wbem_cimom_active_clients">Active CIMOM Clients</Label>
      </Display>
    </ColumnDescriptor>
  </TableDescriptor>
  <QueryDescriptor FETCHLET_ID="Wbem">
    <Property NAME="username" SCOPE="GLOBAL">guest</Property>
    <Property NAME="password" SCOPE="GLOBAL">guest</Property>
    <Property NAME="CIMclassname" SCOPE="GLOBAL">EX_SFLProvider</Property>
    <Property NAME="operation" SCOPE="GLOBAL">COUNT</Property>
  </QueryDescriptor>
</Metric>
```

The FETCHLET\_ID is identified as Wbem. Property names are passed to the fetchlet for the required parameters username, password, and CIMclassname. The operation is identified as COUNT.

The following example shows how to implement a Response Status metric to determine whether the CIMOM is running or not. It returns a value of 1 if the connection to the CIMOM is successful, otherwise 0.

#### Example 7-2 Response Status Metric

```
<Metric NAME="Response" TYPE="TABLE">
  <Display>
    <Label NLSID="wbem_cimom_response">Response</Label>
  </Display>
  <TableDescriptor>
    <ColumnDescriptor NAME="Status" TYPE="NUMBER" IS_KEY="FALSE">
      <Display>
        <Label NLSID="wbem_cimom_response_status">Status</Label>
      </Display>
    </ColumnDescriptor>
  </TableDescriptor>
</Metric>
```

```

        </ColumnDescriptor>
    </TableDescriptor>
    <QueryDescriptor FETCHLET_ID="Wbem">
        <Property NAME="username" SCOPE="GLOBAL">guest</Property>
        <Property NAME="password" SCOPE="GLOBAL">guest</Property>
        <Property NAME="operation" SCOPE="GLOBAL">STATUS</Property>
    </QueryDescriptor>
</Metric>

```

The default operation is the VALUES operation. It is used to fetch the values of a class that is defined in the CIMOM.

In the final example, the EX\_Teacher class is accessed and fetches the name column. Name is the key of the class and of the new metric being defined, so the IS\_KEY property is set to true. The CIM class properties will be mapped to the Enterprise Manager columns in the order that they are specified in the properties property. In this case, there is only 1 property - Name.

### Example 7-3 Single Property Fetched for a Class

```

<Metric NAME="EX_Teacher" TYPE="TABLE">
    <Display>
        <Label NLSID="wbem_EX_Teacher">EX_Teacher Class</Label>
    </Display>
    <TableDescriptor>
        <ColumnDescriptor NAME="Name" TYPE="STRING" IS_KEY="TRUE">
            <Display>
                <Label NLSID="wbem_ex_teacher_name">Name</Label>
            </Display>
        </ColumnDescriptor>
    </TableDescriptor>
    <QueryDescriptor FETCHLET_ID="Wbem">
        <Property NAME="username" SCOPE="GLOBAL">guest</Property>
        <Property NAME="password" SCOPE="GLOBAL">guest</Property>
        <Property NAME="CIMclassname" SCOPE="GLOBAL">EX_Teacher</Property>
        <Property NAME="properties" SCOPE="GLOBAL">Name</Property>
    </QueryDescriptor>
</Metric>

```

If multiple properties are fetched for a class, semi-colons should separate them. The properties should be provided in the order that the column descriptors are specified for the metric table definition.

### Example 7-4 Multiple Properties Fetched for a Class

```

<Metric NAME="EX_SFLProvider" TYPE="TABLE">
    <Display>
        <Label NLSID="wbem_EX_SFLProvider">EX_SFLProvider Class</Label>
    </Display>
    <TableDescriptor>
        <ColumnDescriptor NAME="Name" TYPE="STRING" IS_KEY="TRUE">
            <Display>
                <Label NLSID="wbem_ex_sfl_name">Name</Label>
            </Display>
        </ColumnDescriptor>
        <ColumnDescriptor NAME="Win" TYPE="NUMBER" IS_KEY="FALSE">
            <Display>
                <Label NLSID="wbem_ex_sfl_win">Win</Label>
            </Display>
        </ColumnDescriptor>
    </TableDescriptor>
    <QueryDescriptor FETCHLET_ID="Wbem">
        <Property NAME="username" SCOPE="GLOBAL">guest</Property>
        <Property NAME="password" SCOPE="GLOBAL">guest</Property>
        <Property NAME="CIMclassname" SCOPE="GLOBAL">EX_SFLProvider</Property>
        <Property NAME="properties" SCOPE="GLOBAL">Name;Win</Property>
    </QueryDescriptor>
</Metric>

```

```

<ColumnDescriptor NAME="Lost" TYPE="NUMBER" IS_KEY="FALSE">
  <Display>
    <Label NLSID="wbem_ex_sfl_lost">Lost</Label>
  </Display>
</ColumnDescriptor>
</TableDescriptor>
<QueryDescriptor FETCHLET_ID="Wbem">
  <Property NAME="username" SCOPE="GLOBAL">guest</Property>
  <Property NAME="password" SCOPE="GLOBAL">guest</Property>
  <Property NAME="CIMclassname" SCOPE="GLOBAL">EX_SFLProvider</Property>
  <Property NAME="properties" SCOPE="GLOBAL">Name;Win;Lost</Property>
</QueryDescriptor>
</Metric>

```

## 7.9 JDBC Fetchlet

Call-level interfaces such as JDBC permit external access to SQL database manipulation and update commands. The Java Database Connectivity (JDBC) fetchlet allows you to execute common JDBC commands and obtain their response time for any type of database.

### Input Parameters

**Table 7-15** *JDBC Fetchlet Input Parameters*

Name	Description	Use
Transaction Name	(Standard)	Required.
Beacon Name	(Standard)	Required.
Connect String	<p>Connection string provided by the user. The Connect String must comply with the URL format specified by the vendor of the database to which the user is trying to connect.</p> <p>Examples:</p> <p>Format required by Oracle:</p> <pre>jdbc:oracle:thin:@hostname:port</pre> <p>Format required by MySQL:</p> <pre>jdbc:mysql://hostname:port</pre>	Required.
Class Name String	<p>The driver class name to be used for connections.</p> <p>Example:</p> <pre>oracle.jdbc.driver.OracleDriver</pre> <p>You have two options for configuring the Agent to use the .jar file containing the driver:</p> <ol style="list-style-type: none"> <li>1. Place the .jar file in \$JAVA_HOME/jre/lib/ext. CLASSPATH does not need to be modified.</li> <li>2. Place the .jar file anywhere and update CLASSPATH in emd.properties file with the path to jar. Bounce Agent. This should be scripted and be transparent to user.</li> </ol>	Required.

**Table 7–15 (Cont.) JDBC Fetchlet Input Parameters**

Name	Description	Use
Username	Username to be used when connecting to the database.	Required.
Password	Password to be used when connecting to the database.	Required.
Role	User Role	Required.
Statement	SQL statement to be executed. Use of PL/SQL is possible by using prepareCall() API.	Required.

**Table 7–16 Metric Columns Collected**

Column	Description
Status	Status of the test. Status is 'down' if there is a SQLException generated by the fetchlet.
Total Time	Time required for the fetchlet to execute the test.
Connect Time	Time required for DriverManager.getConnection() to complete.
Prepare Time	Time required for conn.prepareStatement() to complete.
Execute Time	Time required for stmt.executeQuery() to complete.
Fetch Time	Time required for while(rs.next()) { rs.getRow() } to complete.
Close Time	Time required for closing resultset, statement, connection to complete.
Number of rows	Number of rows fetched.
Total time per row	
Fetch time per row	

## 7.10 OJMX/SOAP Fetchlet

The OJMX fetchlet communicates with the JMX agent on the managed J2EE server and performs the specified operations.

### Input Parameters

**Table 7–17 JMX Fetchlet Input Parameters**

Name	Type	Description
metricType	String	Tells the fetchlet that this metric is of type Web Service (GWS).
requestBodyElement	String	Provides the name of the Web Service operation as used in the Web Service request body.
documentType	String	Specifies SOAP encoding. For example: rpc/encoded or doc/literal.
soapVersion	String	Version of SOAP that the fetchlet should use to communicate with the Web Service (default is 1.2)
targetNamespace	String	Target namespace of Web Service.

**Table 7–17 (Cont.) JMX Fetchlet Input Parameters**

<b>Name</b>	<b>Type</b>	<b>Description</b>
ColumnOrder	String []	Comma separated list of XPath paths to pick the column for a resultant Web Service response.
rowData	String []	Comma separated list of XPath prefixes which, when appended to corresponding columnOrder Xpaths, provide the value of the metric column for each row when arrays are returned in the response from a Web Service.
URI	String	URI of Web Service.
soapAction	String	SOAP action for web service from WSDL.
returnType	String	Type of web service invocation return value.
arguments	XML	Values to pass to the Web Service invocation.

**Output**

The OJMX fetchlet returns an object of type MetricResult that contains the information retrieved.





---

---

# Index

## A

---

Adding Reports, 4-1  
adding target types, 2-1  
alert messages, guidelines, 2-18  
Anonymous PL/SQL Block, specifying, 4-15  
Application Service Level Management Views, 6-50  
Availability Status Icon, 4-18

## C

---

Chart Element, 4-19  
Chart Type, 4-19  
collection frequency, 2-18  
collections file, 2-3  
Column Group End Column, 4-17  
Column Group Header, 4-17  
Column Group Start Column, 4-17  
Configuration Files, 2-2  
Configuration Management Views, 6-91  
configuration metrics, 2-17  
Configuration Views, 6-73

## D

---

Database Cluster Views, 6-109  
Default Collection File, creating, 2-12  
default collections file, 2-3  
defining collections, guidelines, 2-18  
Development Guidelines, 2-16  
development guidelines, reports, 4-25  
DMS Fetchlet/Agent Integration Instructions, 7-23  
DTD, 2-1  
dynamic instance properties, 2-17  
Dynamic Monitoring Service, 7-21  
Dynamic Time Selector, 4-24  
Dynamic Validation, 3-2

## E

---

Extensibility Guide, about, 1-5  
Extensions Exchange, 1-3

## F

---

FETCHLET\_ID, 2-9  
fetchlets, 7-1

File Locations, 2-2  
Fill, 4-19

## H

---

Height, 4-20  
Horizontal or Vertical, 4-20  
HTTP Data Fetchlets, 7-26

## I

---

ILINT, 3-1  
ILINT Examples, 3-5  
ILINT Output, 3-3  
ILINT, validation types  
    validation, XML, 3-1  
Information Publisher, 4-1  
Instance Properties, 2-9  
instance-specific collections, 2-13  
Is PL/SQL Statement, 4-14

## J

---

JDBC Fetchlet, 7-35  
Job Views, 6-45

## L

---

Legend Position, 4-20  
lifecycle  
    deployment, 1-3  
    development, 1-3  
lifecycle, Management Plug-in, 1-3  
Linux Patching Views, 6-83

## M

---

Management Plug-in Archive, creating, 2-19  
Management Plug-in Archive, uploading, 2-20  
Management Plug-in Lifecycle, 1-3  
Management Plug-ins, 1-1  
Management Plug-ins Availability, 1-3  
Management Plug-ins, troubleshooting, 2-23  
Management Template Views, 6-40  
Maximum Number of Rows, 4-16  
Metadata File, 2-3

metadata version, checking, 3-2  
 Metadata, definition guidelines, 2-16  
 Metric Browser, 2-10  
 Metric Details Element, 4-22  
 metric evaluation order, 2-18  
 metric naming conventions, 2-17  
 MGMT\$ALERT\_ANNOTATIONS, 6-7  
 MGMT\$ALERT\_CURRENT, 6-15  
 MGMT\$ALERT\_HISTORY, 6-17  
 MGMT\$ALERT\_NOTIF\_LOG, 6-8  
 MGMT\$APPL\_PATCH\_AND\_PATCHSET, 6-82  
 MGMT\$APPLIED\_PATCHES, 6-82  
 MGMT\$APPLIED\_PATCHSETS, 6-83  
 MGMT\$AVAILABILITY\_CURRENT, 6-14  
 MGMT\$AVAILABILITY\_HISTORY, 6-14  
 MGMT\$BLACKOUT\_HISTORY, 6-5  
 MGMT\$BLACKOUTS, 6-6  
 MGMT\$CLUSTER\_INTERCONNECTS, 6-110  
 MGMT\$CSA\_COLLECTIONS, 6-91  
 MGMT\$CSA\_FAILED, 6-95  
 MGMT\$CSA\_HOST\_COOKIES, 6-96  
 MGMT\$CSA\_HOST\_CPUS, 6-97  
 MGMT\$CSA\_HOST\_CUSTOM, 6-97  
 MGMT\$CSA\_HOST\_IUCARDS, 6-98  
 MGMT\$CSA\_HOST\_NICS, 6-98  
 MGMT\$CSA\_HOST\_OS\_COMPONENTS, 6-95  
 MGMT\$CSA\_HOST\_OS\_FILESYSTEMS, 6-99  
 MGMT\$CSA\_HOST\_OS\_PROPERTIES, 6-99  
 MGMT\$CSA\_HOST\_RULES, 6-97  
 MGMT\$CSA\_HOST\_SW, 6-96  
 MGMT\$CSM\_DOMAIN\_DAILY, 6-66  
 MGMT\$CSM\_DOMAIN\_DIST\_DAILY, 6-68  
 MGMT\$CSM\_DOMAIN\_DIST\_HOURLY, 6-67  
 MGMT\$CSM\_DOMAIN\_HOURLY, 6-65  
 MGMT\$CSM\_IP\_DAILY, 6-60  
 MGMT\$CSM\_IP\_DIST\_DAILY, 6-61  
 MGMT\$CSM\_IP\_DIST\_HOURLY, 6-61  
 MGMT\$CSM\_IP\_HOURLY, 6-59  
 MGMT\$CSM\_METRIC\_DETAILS, 6-51  
 MGMT\$CSM\_MT\_IP\_DAILY, 6-63  
 MGMT\$CSM\_MT\_IP\_DIST\_DAILY, 6-65  
 MGMT\$CSM\_MT\_IP\_DIST\_HOURLY, 6-64  
 MGMT\$CSM\_MT\_IP\_HOURLY, 6-62  
 MGMT\$CSM\_MT\_METRIC\_DETAILS, 6-52  
 MGMT\$CSM\_MT\_URL\_DAILY, 6-56  
 MGMT\$CSM\_MT\_URL\_DIST\_DAILY, 6-58  
 MGMT\$CSM\_MT\_URL\_DIST\_HOURLY, 6-57  
 MGMT\$CSM\_MT\_URL\_HOURLY, 6-55  
 MGMT\$CSM\_REGION, 6-50  
 MGMT\$CSM\_SUBNET\_DAILY, 6-69  
 MGMT\$CSM\_SUBNET\_DIST\_DAILY, 6-70  
 MGMT\$CSM\_SUBNET\_DIST\_HOURLY, 6-70  
 MGMT\$CSM\_SUBNET\_HOURLY, 6-68  
 MGMT\$CSM\_URL\_DAILY, 6-53  
 MGMT\$CSM\_URL\_DIST\_DAILY, 6-55  
 MGMT\$CSM\_URL\_DIST\_HOURLY, 6-54  
 MGMT\$CSM\_URL\_HOURLY, 6-52  
 MGMT\$CSM\_WATCHLIST, 6-50  
 MGMT\$DB\_CONTROLFILES, 6-74  
 MGMT\$DB\_DATAFILES, 6-74  
 MGMT\$DB\_DBNINSTANCEINFO, 6-75  
 MGMT\$DB\_FEATUREUSAGE, 6-76  
 MGMT\$DB\_INIT\_PARAMS, 6-77  
 MGMT\$DB\_LICENSE, 6-77  
 MGMT\$DB\_OPTIONS, 6-81  
 MGMT\$DB\_REDOLOGS, 6-78  
 MGMT\$DB\_ROLLBACK\_SEGS, 6-79  
 MGMT\$DB\_SGA, 6-80  
 MGMT\$DB\_TABLESPACES, 6-73, 6-80  
 MGMT\$E2E\_1DAY, 6-71  
 MGMT\$E2E\_HOURLY, 6-71  
 MGMT\$E2E\_RAW, 6-72  
 MGMT\$ECM\_CONFIG\_HISTORY, 6-99  
 MGMT\$ECM\_CONFIG\_HISTORY\_KEY1, 6-100  
 MGMT\$ECM\_CONFIG\_HISTORY\_KEY2, 6-101  
 MGMT\$ECM\_CONFIG\_HISTORY\_KEY3, 6-101  
 MGMT\$ECM\_CONFIG\_HISTORY\_KEY4, 6-102  
 MGMT\$ECM\_CONFIG\_HISTORY\_KEY5, 6-102  
 MGMT\$ECM\_CONFIG\_HISTORY\_KEY6, 6-103  
 MGMT\$EM\_HOMES\_PLATFORM, 6-82  
 MGMT\$ESA\_ALL\_PRIVS\_REPORT, 6-84  
 MGMT\$ESA\_ANY\_DICT\_REPORT, 6-85  
 MGMT\$ESA\_ANY\_PRIV\_REPORT, 6-85  
 MGMT\$ESA\_AUDIT\_SYSTEM\_REPORT, 6-85  
 MGMT\$ESA\_BECOME\_USER\_REPORT, 6-86  
 MGMT\$ESA\_CATALOG\_REPORT, 6-86  
 MGMT\$ESA\_CONN\_PRIV\_REPORT, 6-86  
 MGMT\$ESA\_CREATE\_PRIV\_REPORT, 6-87  
 MGMT\$ESA\_DBA\_GROUP\_REPORT, 6-87  
 MGMT\$ESA\_DBA\_ROLE\_REPORT, 6-87  
 MGMT\$ESA\_DIRECT\_PRIV\_REPORT, 6-88  
 MGMT\$ESA\_EXMPT\_ACCESS\_REPORT, 6-88  
 MGMT\$ESA\_KEY\_OBJECTS\_REPORT, 6-88  
 MGMT\$ESA\_OH\_OWNERSHIP\_REPORT, 6-89  
 MGMT\$ESA\_OH\_PERMISSION\_REPORT, 6-89  
 MGMT\$ESA\_POWER\_PRIV\_REPORT, 6-89  
 MGMT\$ESA\_PUB\_PRIV\_REPORT, 6-90  
 MGMT\$ESA\_SYS\_PUB\_PKG\_REPORT, 6-90  
 MGMT\$ESA\_TABSP\_OWNERS\_REPORT, 6-90  
 MGMT\$ESA\_TRC\_AUD\_PERM\_REPORT, 6-90  
 MGMT\$ESA\_WITH\_ADMIN\_REPORT, 6-91  
 MGMT\$ESA\_WITH\_GRANT\_REPORT, 6-91  
 MGMT\$HA\_BACKUP, 6-111  
 MGMT\$HOMES\_AFFECTED, 6-82  
 MGMT\$HOSTPATCH\_GROUPS, 6-84  
 MGMT\$HOSTPATCH\_GRP\_COMPL\_HIST, 6-84  
 MGMT\$HOSTPATCH\_HOST\_COMPL, 6-84  
 MGMT\$HOSTPATCH\_HOSTS, 6-83  
 MGMT\$HW\_NIC, 6-104  
 MGMT\$JOB\_ANNOTATIONS, 6-48  
 MGMT\$JOB\_EXECUTION\_HISTORY, 6-47  
 MGMT\$JOB\_NOTIFICATION\_LOG, 6-49  
 MGMT\$JOB\_STEP\_HISTORY, 6-47  
 MGMT\$JOB\_TARGETS, 6-46  
 MGMT\$JOBS, 6-45  
 MGMT\$METRIC\_CATEGORIES, 6-30  
 MGMT\$METRIC\_CURRENT, 6-20  
 MGMT\$METRIC\_DAILY, 6-23  
 MGMT\$METRIC\_DETAILS, 6-19  
 MGMT\$METRIC\_ERROR\_CURRENT, 6-3

MGMT\$METRIC\_ERROR\_HISTORY, 6-3  
MGMT\$METRIC\_HOURLY, 6-22  
MGMT\$OS\_COMPONENTS, 6-104  
MGMT\$OS\_FS\_MOUNT, 6-104  
MGMT\$OS\_HW\_SUMMARY, 6-105  
MGMT\$OS\_KERNEL\_PARAMS, 6-105  
MGMT\$OS\_PATCHES, 6-105  
MGMT\$OS\_SUMMARY, 6-106  
MGMT\$POLICIES, 6-30  
MGMT\$POLICY\_PARAMETERS, 6-32  
MGMT\$POLICY\_VIOL\_ANNOTATIONS, 6-33  
MGMT\$POLICY\_VIOL\_NOTIF\_LOG, 6-34  
MGMT\$POLICY\_VIOLATION\_CONTEXT, 6-40  
MGMT\$POLICY\_VIOLATION\_CTXT, 6-32  
MGMT\$POLICY\_VIOLATION\_CURRENT, 6-37  
MGMT\$POLICY\_VIOLATION\_HISTORY, 6-39  
MGMT\$RACDB\_INTERCONNECTS, 6-110  
MGMT\$SOFTWARE\_COMP\_PATCHSET, 6-106  
MGMT\$SOFTWARE\_COMPONENT\_ONEOFF, 6-106  
MGMT\$SOFTWARE\_COMPONENTS, 6-107  
MGMT\$SOFTWARE\_DEPENDENCIES, 6-107  
MGMT\$SOFTWARE\_HOMES, 6-108  
MGMT\$SOFTWARE\_ONEOFF\_PATCHES, 6-108  
MGMT\$SOFTWARE\_OTHERS, 6-108  
MGMT\$SOFTWARE\_PATCHES\_IN\_HOMES, 6-109  
MGMT\$SOFTWARE\_PATCHSETS, 6-109  
MGMT\$STORAGE\_REPORT\_DATA, 6-112  
MGMT\$STORAGE\_REPORT\_DISK, 6-114  
MGMT\$STORAGE\_REPORT\_ISSUES, 6-114  
MGMT\$STORAGE\_REPORT\_KEYS, 6-113  
MGMT\$STORAGE\_REPORT\_LOCALFS, 6-115  
MGMT\$STORAGE\_REPORT\_NFS, 6-116  
MGMT\$STORAGE\_REPORT\_PATHS, 6-113  
MGMT\$STORAGE\_REPORT\_VOLUME, 6-115  
MGMT\$TARGET, 6-24  
MGMT\$TARGET\_ASSOCIATIONS, 6-27  
MGMT\$TARGET\_COMPONENTS, 6-4  
MGMT\$TARGET\_FLAT\_MEMBERS, 6-28  
MGMT\$TARGET\_MEMBERS, 6-28  
MGMT\$TARGET\_METRIC\_COLLECTIONS, 6-10  
MGMT\$TARGET\_METRIC\_SETTINGS, 6-11  
MGMT\$TARGET\_POLICIES, 6-35  
MGMT\$TARGET\_POLICY\_EVAL\_SUMM, 6-33  
MGMT\$TARGET\_POLICY\_SETTINGS, 6-36  
MGMT\$TARGET\_PROPERTIES, 6-29  
MGMT\$TARGET\_TYPE, 6-26  
MGMT\$TARGET\_TYPE\_DEF, 6-27  
MGMT\$TARGET\_TYPE\_PROPERTIES, 6-29  
MGMT\$TEMPLATE\_METRIC\_SETTINGS, 6-43  
MGMT\$TEMPLATE\_METRICCOLLECTION, 6-42  
MGMT\$TEMPLATE\_POLICY\_SETTINGS, 6-41  
MGMT\$TEMPLATES, 6-41  
MGMT\_IP\_PARAM\_VALUE\_LIST, 4-12  
MGMT\_IP\_PARAM\_VALUE\_RECORD, 4-12  
MGMT\_IP\_TARGET\_TYPES, 4-12  
mgmt\_ip.add\_element\_to\_report\_def, 4-10  
mgmt\_ip.create\_report\_definition, 4-9  
mgmt\_mp\_homepage.add\_report, 4-11  
Monitoring Views, 6-3

## N

---

Name Value Pair Display, 4-14  
named notation, reports, 4-27  
Null Data String Substitute, 4-16  
Number of Rows to Show, 4-14  
number of rows, controlling, 2-18

## O

---

OJMX/SOAP Fetchlet, 7-36  
Oracle Home Patching Views, 6-81  
Oracle Technology Network, 1-3  
OS Command Fetchlets, 7-1  
OSFetchlet, 7-1  
OSLinesFetchlet, 7-3  
OSLineTokenFetchlet, 7-5

## P

---

performance metrics, 2-17  
PL/SQL Type Definitions, 4-12  
Policy Violation Views, 6-37

## Q

---

Query Descriptor, 2-9

## R

---

Related Links, adding, 2-14  
report definition file, 4-3  
report definition files, creating, 4-4  
Report Definitions Page, 4-2  
report definitions, updating, 4-9  
report testing, interactive, 4-5  
reporting pl/sql api, 4-5  
Reports Page, 4-1  
Report-Wide Parameters, 4-24  
Repository Views, 6-1  
repository views, 6-1

## S

---

SCOPE, 2-9  
scripts and binaries, plug-in, 2-15  
Security Views, 6-84  
Security, Web Services, 5-2  
Separate Rows as Delimiters, 4-18  
Separate Rows for Values in Cell, 4-17  
Severity Icon, 4-18  
SNMP Fetchlet, 7-12  
SOAP, 5-1  
Sort Column, 4-13  
Sort Order, 4-14  
Split Table into Multiple Tables by Column, 4-17  
SQL Fetchlet, 7-7  
SQL or PL/SQL queries, reports, 4-5  
SQL or PL/SQL Statement, 4-14  
static instance properties, 2-16  
static validation, 3-1

Storage Reporting Views, 6-112  
SYSTEM Reports, adding, 2-14

## T

---

Table Element Parameters, 4-13  
target definition, 2-3  
Target Definition File, anatomy, 2-4  
Target Definition Files, 2-3  
target descriptors  
    Metric NAME, 2-8  
    TargetMetadata and Display, 2-7  
Target Home Page, 1-2  
target instance, adding, 2-22  
Target Type, 4-18  
target type metadata, 2-3  
    anatomy of, 2-4  
    file naming conventions, 2-4  
target type validation, 2-10  
target types, adding, 2-1  
target, adding, 2-1  
TargetInstance.dtd, 2-3  
TargetMetadata.dtd, 2-3  
targets.xml, adding target instances, 2-11  
Text Element Parameters, 4-24  
Time Period, 4-13, 4-19

## U

---

URL Fetchlet (raw), 7-26  
URL Line Token Fetchlet, 7-28  
URL Lines Fetchlet, 7-27  
URL Timing Fetchlet, 7-15  
URLXML Fetchlet, 7-29

## V

---

Validation, XML, 3-1

## W

---

Web Service, generating files, 5-2  
Web Services, 5-1  
Web Services Command Line Tool, 5-2  
Web Services, monitoring, 5-1  
WSDL, 5-1

## X

---

XML, 2-1  
XML, testing, 2-15