Hyperformix®
a CA Technologies company

Custom Data Adapter SDK™ 2.0
for Data Manager 4.3
Reference Guide

May, 2011

Hyperformix, Inc.
a CA Technologies company
5001 Plaza on the Lake Blvd, Suite 102
Austin, TX 78746

Contents

1.0 Introduction

The Hyperformix® Custom Data Adapter Software Development Kit (SDK) provides a set of example files and instructions to create custom Data Adapters for use with Hyperformix Data Manager®. Custom Data Adapters are required to gather data from data sources other than those directly supported by Hyperformix. This document provides the information necessary to create and install custom Data Adapters.

1.1 About Data Manager and Data Adapters

Data Manager is a data warehousing tool that collects and stores various kinds of data, such as performance metrics, configuration data, workload, and transaction metrics. Data Manager collects this data from system management tools from vendors such as HP®, Tivoli®, VMware®, and Microsoft®. The source data can reside in a relational database such as Oracle®, a text file, or be accessed through an API such as web services. Data Manager connects to these data sources to collect, normalize, and store the data in the Data Manager database. You can then use the Data Manager database to create a variety of reports or as a data source for other Hyperformix products, such as Capacity Manager® or Performance Optimizer®.

The Data Manager component responsible for gathering the data is called a Data Adapter. The Data Manager product is installed with a number of standard Data Adapters, and additional Data Adapters are available in the Hyperformix Solution Kits.

To develop a custom Data Adapter, you must create an XML file in the required format that conforms to the Data Manager Data Adapter schema and you must have data available in either a database or in a correctly formatted Comma Separated Value (CSV) text file. After you have created and successfully imported a custom Data Adapter, you can use it as you would a standard Data Adapter to perform Data Manager operations such as creating data sources, loading data, migrating data, and creating reports.

1.2 Prerequisites

To create a custom Data Adapter, you create an XML file that specifies the form of the data source (relational database or flat file), what data to obtain from the source and how to obtain it, and where the data is stored in the Data Manager database.

To do this, you must

- be familiar with Hyperformix Data Manager concepts
- know the purpose of the Data Adapter

  For example, you must know whether the Data Adapter will be used to gather data for modeling in Hyperformix Capacity Manager, Hyperformix Performance Optimizer®, or for reporting purposes.
- know what data is required to accomplish the intended purpose

  For example, if you are going to use the data in Capacity Manager, you should know what data needed (and not needed) to model a system.
- understand how the required data is stored in the data source

  This includes knowing what data is available, the units associated with the data, and location of the data within the source.
- be familiar with XML concepts and coding

  You must be able to edit an XML file and understand any errors that might occur when the Data Adapter is validated against the Data Manager Data Adapter schema.
- be familiar with SQL and database concepts (such as tables, views, and synonyms) (when working with database data sources)

  You must have the authorizations and the knowledge to write queries necessary to obtain the data from the source database.

1.3 About the example files

In addition to this document, the Custom Data Adapter SDK includes template and example Data Adapter files, example data, and the Data Manager XML schema and schema documentation. You can use these files when going through the examples in this document and as templates when creating your own custom Data Adapters.

1.3.1 Examples

The **\examples** directory in the Custom Data Adapter SDK .zip file contains four subdirectories, each containing a template Data Adapter XML file, sample custom Data Adapter files, and sample data (if applicable). This document explains how to create custom Data Adapters using these samples as a point of reference.

You can use these files as a starting point when creating new Data Adapters.

1.3.1.1 Examples for CSV data sources

The **\examples\da_csv** directory contains Data Adapter XML files and sample CSV source data files.

Table 1: Example files for Data Adapters for CSV data

File name	Description
generic_csv_template.xml	A Data Adapter XML file for CSV source data which contains maps for all supported data files (contained in the /sample_data folder). This file can be used as a template when creating custom Data Adapters for CSV source data.
server_config_and_disk_da.xml	A Data Adapter XML file for CSV source data which contains maps for physical server configuration and disk data. This sample file is described in the exercise in *5.4 Developing a sample CSV Data Adapter*.
virtual_server_config_and_disk_da.xml	A Data Adapter XML file for CSV source data which contains maps for virtual machine configuration and disk data.
group_by_country_da.xml	A Data Adapter XML file for CSV source data which contains a map for a hierarchical logical grouping of country, state, city.

File name	Description
Generic_Cluster_Configuration.csv Generic_Cluster_Metrics.csv Generic_Resourcepool_Metrics.csv Generic_respool_configuration.csv Generic_transaction.csv Generic_workload.csv GenericApplication.csv GenericApplication_virtual.csv GenericConfiguration.csv GenericConfiguration_virtual.csv GenericCPU.csv GenericCPU_virtual.csv GenericDisk.csv GenericDisk_virtual.csv GenericNIC.csv GenericNIC_virtual.csv GenericProcess.csv GenericProcess_virtual.csv GenericSystem.csv GenericSystem_virtual.csv GroupByBusinessService.csv GroupByCountry.csv GroupByDatacenter.csv	Sample data files containing metrics which can be collected into Data Manager. The generic_csv_template.xml is defined using the data from these files.

1.3.1.2 Examples for database data sources

The **\examples\da_db** directory contains Data Adapter XML files.

File name	Description
db_template.xml	A Data Adapter XML file for database source data which contains maps for all supported data source types. This file can be used as a template when creating custom Data Adapters for database source data.
hyperformix_datamanager_43_da.xml	A Data Adapter XML file which contains database queries for server configuration and metrics. The "43" in the XML name indicates it is valid with the Data Manager 4.3 schema. This is a working Data Adapter which can be used to obtain data from a Data Manager database.

1.3.1.3 Examples of diverse data source files

The **\examples\da_diverse_data** directory contains an example Data Adapter and example diverse data source files. You can model your own custom Data Adapter and diverse data sources on these files.

File name	Description
diverse_data_example_da.xml	An example Data Adapter XML file for diverse data source files.
TCO_data.txt	A source file containing example server configuration data.
users_data.txt	A source file containing example user configuration data.

1.3.1.4 Examples of diverse metric source files

The **\examples\da_diverse_metric** directory contains sample Data Adapters and diverse metrics source files. You can model your own Data Adapters and diverse metrics sources on these files.

File name	Description
diverse_metric_san_da.xml	An example Data Adapter XML file for diverse metric source file which contains a map for the SAN metrics contained in the **san_data.txt** file.
san_data.txt	A source data file containing SAN metrics. This is the source data used by the **diverse_metric_san_da.xml** example Data Adapter.
diverse_metric_keynote_response_time_da.xml	An example Data Adapter XML file for diverse metric source file which contains a map for the Keynote® response time metrics contained in the **keynote_response_time_data.txt** file.
keynote_response_time_data.txt	A source data file containing Keynote response time metrics. This is the source data used by the **diverse_metric_keynote_response_time_da.xml**

1.3.2 XML Schema

This directory contains the XML schema for a Data Adapter and the html files which provide a visual navigation of the schema. For more information, see *3.1 About the XML schema*

The **\xml_schema** directory contains XML schema files and documentation

File name	Description
DST_4_3.xsd	Data Adapter XML schema file. The DA XML files must validate against this schema.
xml_schema\html_docs*	HTML documentation files providing a visual representation of the schema.

2.0 Supported data source metrics

Custom Data Adapters can collect data from standard relational databases such as Oracle, DB2 and Microsoft SQL server, and from Comma Separated Values (CSV) files.

The following data source metric classes are supported:

- Physical server configuration and performance metrics
- Virtual server configuration and performance metrics
- Cluster configuration and performance metrics
- Resource pool configuration and performance metrics
- Workload level metrics
- Transaction level metrics
- Diverse metrics
- Diverse data

Note: The diverse data and diverse metrics classes allow Data Manager to store data and metrics (such as energy utilization metrics in the data center) that do not fall into the other classes. Refer to the *Data Manager Administration Guide* for more information.

The following tables show the metric groups and the staging tables in which they are stored in the Data Manager database for each metric class. It also maps the metric group to the custom Data Adapter XML file destination element. Mapping details are provided in *10.0 XML destination details*.

2.1 Physical server metric class

Metric group	Description	Staging table	XML destination element
System Metrics	System level metric information such as CPU, memory, disk, network utilization.	server_metrics_ staging	server_metrics
CPU metrics	Per CPU metrics such as CPU utilization and wait time.	cpu_metrics_staging	cpu_metrics
Disk Metrics	Per physical disk metrics such as disk utilization and read and write rates.	disk_metrics_staging	disk_metrics
Network Metrics	Per NIC metrics such as bytes in/out rate and total bandwidth.	network_metrics_ staging	network_metrics
Process Metrics	Per process metrics such as process CPU and memory utilization.	process_metrics_ staging	process_metrics
Application Metrics	Per application metrics such as application CPU and memory utilization.	application_metrics_ staging	application_metrics
Server Configuration Data	Configuration data of the physical server such as the number of CPUs and total memory.	server_staging	server

2.2 Virtual machine metric class

Metric group	Description	Staging table	XML destination element
System Metrics	System level metric information such as CPU, memory, disk, and network utilization.	server_metrics_ staging	server_metrics
CPU metrics	Per CPU metrics such as CPU utilization and wait time.	cpu_metrics_staging	cpu_metrics
Disk Metrics	Per physical disk metrics such as disk utilization and read and write rate.	disk_metrics_staging	disk_metrics
Network Metrics	Per NIC metrics such as bytes in/out rate and total bandwidth.	network_metrics_ staging	network_metrics
Process Metrics	Per process metrics such as process CPU and memory utilization.	process_metrics_ staging	process_metrics
Application Metrics	Per application metrics such as application CPU and memory utilization.	application_metrics_ staging	application_metrics
Virtual Machine Configuration	Configuration data of the VM such as the number of virtual CPUs, total memory, cluster/resource pool to which the VM belongs, CPU and memory entitlements/shares.	server_virtual_staging	server_virtual

2.3 Cluster metric class

Metric group	Description	Staging table	XML destination element
Cluster Performance Metrics	Cluster performance metrics such as CPU and memory utilization of the cluster.	cluster_metrics_ staging	cluster_metrics
Cluster Configuration	Configuration data such as whether the cluster is enabled for load balancing, high availability, and total computing power.	cluster_staging	cluster

2.4 Resource pool metric class

Metric group	Description	Staging table	XML destination element
Resource Pool performance Metrics	Resource pool performance metrics such as CPU and memory utilization of the resource pool.	resource_pool_ metrics_ staging	resource_pool_ metrics
Resource Pool Configuration	Configuration data such as CPU and memory resources available in the pool.	resource_pool_staging	resource_pool

2.5 Workload data metric class

Metric group	Description	Staging table	XML destination element
Workload Metrics	Metrics such as response time, visit count for a business process, IT service, and business function.	workload_log_staging	workload_log

2.6 Transactional data metric class

Metric group	Description	Staging table	XML destination element
Transaction Metrics	Metrics such as total CPU seconds used by the transaction and memory consumed.	work_trans_metrics_ staging	work_trans_ metrics

2.7 Diverse metrics metric class

Metric group	Description	Staging table	XML destination element
Diverse Metrics	Metric data that does not fall into the other categories (for example quarterly sales data, energy utilization metrics in the data center).	diverse_metrics_ staging	diverse_metrics

2.8 Diverse data metric class

Metric group	Description	Staging table	XML destination element
Diverse Data	Arbitrary data not in another category.	diverse_data_staging	diverse_data

2.9 Group configuration metric class

Metric group	Description	Staging table	XML destination element
Group Configuration	Groupings of servers, VMs, clusters, workloads, or transactions into hierarchies (for example, Country, State, City).	group_config_staging	group_config

3.0 About the Data Adapter XML file

You create a custom Data Adapter by developing an XML file that specifies the following:

- the type of the data source (relational database or flat file)
- the data you want to obtain from the source and how to obtain it
- where the data is to be stored in the Data Manager database

The Custom Data Adapter SDK includes the following sample Data Adapters, which you can use as templates when creating your custom Data Adapter:

- **db_template.xml** - a sample Data Adapter XML file for a database source.
- **generic_csv_template.xml** - a sample Data Adapter XML file for CSV files source.

3.1 About the XML schema

The Data Adapter XML file must conform to the schema defined in the **DST_4_3.xsd** schema file. This schema file is included in the Custom Data Adapter SDK along with an HTML documentation set which provides a visual navigation tool for the schema. To view the schema definition using this tool, unzip the **DST_4_3_Docs.zip** included in this kit, and open the **DST_4_3_Docs.html** file in a browser.

3.2 XML elements

The following section describes the elements of the Data Adapter XML file.

data_source_type

Identifies the Data Manager schema. Do not change this element.

name

The name for the data source type associated with the Data Adapter.

You should specify a brief but descriptive name. This name is displayed in the Data Manager data source type lists (for example, in the Manage Source Types and Manage Data Sources dialogs).

author

The author of the Data Adapter. The author is included in the Overview section of the Data Source Type Details report.

copyright

Copyright information. The copyright is included in the Overview section of the Data Source Type Details report.

description

A brief description of the data source type. This description should include information that Data Manager users should know when using the data source type. It is displayed in the Overview section of the Data Source Type Details report.

> **Note**: Use a CDATA section for the description to avoid needing to use escape sequences for XML special characters. HTML markup (such as lists and tables) is allowed.

data_aquisition_class

The name of the Java class that gathers the data. You should always choose one of the following classes depending on which data source you are using:

- **ITM_DB_Engine** (for database sources)
- **Common_CSV_Engine** (for CSV sources)

server_list_query

(*database source only*) The SQL select statement used to obtain the list of servers when creating data sources.

3.3 XML map entries

Each map entry has a source element which describes data in the data source and a destinations element which describes how the source data maps to Data Manager metrics, tables and columns.

Each map entry represents a single metric group (system, process, disk, network, and so on) and contains the following elements:

- Map ID
- Source Elements
- Select Elements
- Column Elements
- Destination Elements

3.3.1 Map ID

Each map entry is identified by a the <id> attribute, which is a unique string. Hyperformix recommends choosing ids that reflect the purpose of the map (for example, “processMetricsMap”).

3.3.2 Source element

The <source> element defines how data is obtained from the data source. It consists of zero or more <select> elements and one or more <column> elements.

The following is an example of a source element:

```
<source>

     <select>select server_name, time_period, Duration,
     GBL_CPU_TOTAL_UTIL from server_KERNEL where time_
     period BETWEEN DGE_FROM_DATETIME AND DGE_TO_
     DATETIME AND server_name IN (DGE_SERVERS_LIST)
     </select>
     <column name="server_name" type="string"/>
     <column name="time_period" type="sql_date_time"/>
     <column name="Duration" type="string"/>
     <column name="GBL_CPU_TOTAL_UTIL" type="string"/>

</source>
```

3.3.3 Select elements

If the Data Adapter is for a database source, then the <source> element must contain a <select> element that contains the SQL statement required to get the data from the source database.

Note: The <select> element is not applicable for Data Adapters for CSV file data sources.

3.3.4 Column elements

The <column> elements describe the data gathered from the data source, as follows:

- If the Data Adapter is for a database source, then the <column> elements represent the columns returned by the SQL statement.
- If the Data Adapter is for a CSV file, then the <column> elements represent the column headers in the CSV file.

The attributes associated with the <column> element are described in the following table:

Table 1: Attributes for column elements

Attribute name	Description	Values	Applicable to database source	Applicable to CSV file source	Required
name	Name of the column. For database sources, the name of a column returned by the SQL statement. For CSV sources, the name of a column in the file header.	Name of the column	Yes	Yes	Yes
type	Data type of the column. For allowed values, see Allowed Data Types	Various	Yes	Yes	Yes
date_ format	Format of the date string in the source.	Java date/time format string	Yes	Yes	Required if the data type is date.
optional	A boolean flag indicating whether the column is optional in the data source. If set to **true**, the column may or may not be present in the CSV file.	true or false	No	Yes	No
alias	Alias name for a column. This is useful if the same column is referenced by two different names in different files. For additional information see Alias Names for Source Columns.	Alias name for a column	No	Yes	No

3.3.4.1 Alias names for source columns

If the same column is referenced by different names in different files, you can specify them using the alias name feature of the XML DA definition. Alias names apply to Data Adapters with a CSV data source only, and the Data Adapters treat the name and the alias name attributes identically.

- If more than one alias name is required, define the alias element as a child element of the column.

  The following is an example showing both “Server Name” and "Host Name" defined as aliases for “Server_Name”.

```
<column name="Server_Name"  type="string">
    <alias name="Server Name" />
    <alias name="Host Name" />
</column>
```

  For more details, see the **generic_csv_template.xml** included in the Custom Data Adapter SDK.

3.3.4.2 Space and special characters in source column name

In the Data Adapter XML file, special characters and spaces are not allowed for the column names under the source element. If the source column names contain these characters, perform one of the following steps to change them:

- If the Data Adapter is written for a database, then create an alias for the name within the SQL select query. For example

  ```
  Select "server name" as server_name from table_A;
  ```

- If the Data Adapter is written for a CSV file data source, then create an alias for the name using the alias name feature of the XML DA definition.

  For example if "Server Name" is the name of a column in a CSV file, then you must create an alias for the name as follows:

  ```
  <column name="Server_Name"  type="string">
      <alias name="Server Name" />
  </column>
  ```

3.3.4.3 Allowed data types for column elements

The following table describes the data types allowed in column elements:

Table 2: Allowed data types for column element attributes

Data type	Description	Comments
string	Data is represented as a character string .	---
integer	Data is represented as an integer.	---
double	Data is represented as double(float).	---
date	The data is represented as a date string. In databases, this means that the source database stores the date in a string column instead of storing it in a date, date/time, or timestamp format.	Always specify the format of the date string using the date_ format attribute in the column element.
sql_ date_ time	The date is stored in a date, date/time, or timestamp format in the source database.	Applicable only for database sources.
seconds_ since_ epoch	The date is stored as a number which represents seconds elapsed since epoch (00:00:00 UTC on January 1, 1970)	---

Note: For additional information about Java date and time formats used in the source columns, visit the site: http://java.sun.com/j2se/1.5.0/docs/api/java/text/SimpleDateFormat.html

3.3.5 Destinations element

The <destinations> element describes how the source data maps to the Data Manager metrics, database tables, and columns. Destinations are common and identical for both database and CSV file Data Adapters.

```
<destinations>
  <server_metrics>
      <required_fields>
         <server_id source="server_name"/>
         <metric_time source="time_period"/>
         <gmt_offset source="DGE_GMT_OFFSET"/>
      </required_fields>
      <optional_fields>
         <duration source="Duration"/>
      </optional_fields>
      <metrics>
         <metric name="GBL_CPU_PROC_QUE_LEN" unit="processes"
         source="GBL_RUN_QUEUE"/>
         <metric name="GBL_MEM_UTIL" unit="percent" source="GBL_MEM_UTIL"/>
         <metric name="GBL_CPU_UTIL" unit="percent" source="GBL_CPU_TOTAL_UTIL"/>
      </metrics>
  </server_metrics>
</destinations>
```

3.3.5.1 Attributes for XML Destinations

The <destinations> element describes how the source data maps to the Data Manager metrics, database tables, and columns. You must define a specific set of destinations to import data that belong to different metric groups.

For more information about the metric groups and their corresponding XML destinations, see *2.0 Supported data source metrics*.

Metric Group	XML destination
System Metrics	server_metrics
CPU metrics	cpu_metrics
Disk Metrics	disk_metrics
Network Metrics	network_metrics
Process Metrics	process_metrics
Application Metrics	application_metrics
Server Configuration Data	server
Virtual Machine Configuration	server_virtual
Cluster performance Metrics	cluster_metrics
Cluster Configuration	cluster
Resource Pool performance	resource_pool_metrics
Resource Pool Configuration	resource_pool

Metric Group	XML destination
Workload Data	workload_log
Transactional Data	work_trans_metrics
Diverse Metrics	diverse_metrics
Group Configuration	group_config

Note: It is critical to choose the right destination for a given metric group while developing a custom Data Adapter. If you specify an incorrect destination, the data will be misrepresented in the database.

For example, if the <source> element of a map specifies CPU data but the <destinations> element specifies a disk_metrics destination, the CPU data will be stored in the disk table.

Note: If you want to store data from different source tables in the same destination table, you can create multiple maps for the same destination.

For example, to collect CPU utilization from one table and Memory utilization from another table, you must write separate SQL statements. Each SQL statement specifies a different source, but both would map to the same destination table ("server_metrics").

3.3.5.2 Destination child elements

Destination elements have the following three types of child elements:

required_fields

The members of this group must be present for a given destination when you define a Data Adapter. Required fields differ for various destinations. The names of the required fields for a destination are the names of the columns of the database tables in which data will be stored for this destination.

optional_fields

The members of this group are optional for a given destination when you define a Data Adapter. Optional fields differ for various destinations. The names of the optional fields for a destination are the names of the columns of the database tables in which data will be stored for this destination.

Metrics

The members of this group define metrics to be stored in the corresponding database table. This element contains a list of elements named “metric”. The element “metric” has a few attributes which will be explained below in a table. Metrics are stored as “Metric Name” and “Metric Value” pairs in the database. The “metrics” element may not exist in all destinations and does not exist in destinations that are defined for getting configuration data only such as “server” and “cluster”.

Groups

Group elements specify group names, grouping hierarchy, and the assignment of entities (servers, VMs, clusters, workloads, and so on) to the named groups.

You can specify one or more single groups with a single group element having a name and a source attribute as follows:

```
<group name=”<group name>” source=”<source column>”/>
```

Configuring a group hierarchy is done by nesting the group elements. An example for specifying a group hierarchy of Country, State, and City is:

```
<group name=”Country” source=”<country source column>”
   <group name=”State” source=”<state source column>”
       <group name=”City” source=”<city source column>/>
   </group>
</group>
```

There is no limit on the depth of the group hierarchy.

3.3.5.3 Common attributes for all destination elements

The following attributes are common to all destination elements.

Table 3: Common attributes for destination elements

Attribute Name	Description
source	The source of data for a database column or a metric. **Note**: If the alias names are specified for columns in the <source> element, specify the column name here and not the alias name(s). One of the following: 1. The name of a "column" element defined in the "source" element. 2. An arithmetic expression that may or may not involve names of one or more "column" element defined in the "source" element. For example `(col_A + col_B)/100` 3. A predefined macro. Details on macros will be explained in a later section.

3.3.5.4 Attributes for "metric" elements

The following attributes are specific to the "metric" element of the destination.

Table 4: Attributes for "metric" elements

Attribute Name	Description
name	Name of the metric as it should be stored in the database. For a list of standard predefined names, see *10.0 XML destination details*. You can also define your own metrics
unit	Unit of the metric. When using standard metrics, this should match the predefined units. For the user-defined metrics, you can specify the unit.
description	Description of the metric as it should be stored in the database. For user-defined metrics, choose a description that accurately reflects the metric. This attribute is not required for standard metrics.

3.4 Data Adapter macros

Macros are used to replace absolute values for certain filtering and time related options and allow the user to specify these values from within the Data Manager user interface.

The following Data Adapter macros can be used in the Data Adapter XML files.

Table 1: Data Adapter macros

Macro	Description	When to use
DGE_GMT_OFFSET	GMT offset is a required element in all the destinations that store metric data. When this value is not available from the data source, Data Manager needs to obtain this information from the user through the GUI. This macro enables the Data Adapter to provide Data Manager with the GMT offset information.	When the GMT offset information is not available in the source data, use this macro as the source for the required column "gmt_ offset" in the corresponding destination in the Data Adapter XML file. This is applicable for both database and CSV file Data Adapters.
DGE_FROM_DATETIME	For database sources, you must specify from and to times and a list of servers for which to gather data. This macro is used to filter data based on time. The user-supplied start date/time information is substituted for this Macro at run time while gathering data from the source.	Use this macro in the where clause of the select SQL query for "start time" to filter data based on date and time. This macro is applicable only for database Data Adapters.
DGE_TO_DATETIME	For database sources, you must specify from and to times and a list of servers for which to gather data. This macro is used to filter data based on time. The user-supplied end date/time information is substituted for this Macro at run time while gathering data from the source.	Use this macro in the where clause of the select SQL query for "end time" to filter data based on date and time. This macro is applicable only for database Data Adapters.
DGE_SERVERS_LIST	For database sources, you must specify a list of servers for which to gather data. This macro is used for filtering data based on a list of servers. The user-supplied list of servers is substituted for this macro at run time while gathering data from the source.	Use this macro in the where clause of the select SQL query for "list of servers" to filter data based on list of servers. This macro is applicable only for database Data Adapters.

3.5 Data transformations

Sometimes source data must be transformed before it can be stored in the Data Manager database. For example, you might need to add two source columns or multiply a column by a number. You can perform a transformation using one of the following methods:

1. Specify the arithmetic expression for the transformation as the value of the <source> attribute for a destination element.

   For example, in the metric element below, the value for the metric GBL_CPU_UTIL is derived by using the given arithmetic expression.

   ```
   <metric name="GBL_CPU_UTIL" unit="percent" source="(perc_usr + perc_sys) * 100"/>
   ```

   Note: If the alias names are specified for columns in the <source> element, specify the column name in the arithmetic expression and not the alias name(s).

2. For database sources, you can include transformations in the select statement. Be aware that such transformations will not be evident in the Data Source Type Details report which is generated after the data is imported.

3.6 Data Adapter XML definition files

Before developing a Data Adapter for either a CSV file or a database, you must understand the source data, including the available metrics, their units, and any data transformations required. The XML definition, macros, and process for creation differ depending on the data source type.

The definition files for CSV or database data sources contain the following differences:

Table 2: Differences in XML definition

XML entity	Database DA	CSV DA
"data_aquisition_class" element	Value for this element is "ITM_DB_Engine"	Value for this element is "Common_CSV_Engine"
"server_list_query" element	This is a required element to display the list of servers on the UI. The value is a select query to obtain the list of servers.	This element is not applicable.
"select" element for the source columns	This is a required element to gather data from the source. The value is a select query to obtain the data from the source.	This element is not applicable.
"optional" attribute for the source columns	Not applicable	Applicable to CSV Data Adapters.
"alias" element for the source columns	Not applicable	Applicable to CSV Data Adapters.

4.0 Best practices

Before you begin, familiarize yourself with the following list of best practices to follow when developing a custom Data Adapter.

1. Always develop a Data Adapter on a test or development system. Do not develop it directly on the production system.
2. Always develop a Data Adapter in multiple iterations

   For more information, see *4.1 Developing a Data Adapter using multiple iterations*.
3. When naming a Data Adapter using the **<name>** element, include version information as part of the name.

   Doing so enables you to easily verify that users have the latest version.
4. For database Data Adapters, minimize the use of the **as** clause in SQL statements.

   For more information, see *7.0 Developing a Data Adapter for a database source*.
5. For database Data Adapters, consider filtering multi-instance data (such as process and disk data) based on a threshold value in the **where** clause of the SQL statement.
6. For CSV Data Adapters, when defining multiple maps in the same XML file, ensure that no two map entries use the same set of non-optional columns in the source element.

   Data Manager identifies a map element by comparing the column header in the CSV file with the non-optional columns (columns with no optional attribute or with the optional attribute set to false) in the source element. If you require two map entries for the same set of non-optional columns, you should create two Data Adapters.
7. Always include detailed descriptions for Data Adapters and all user-defined metrics.
8. To ensure that your Data Adapter is working as you expect, always check for Data Adapter details, warnings, and so on when you import a Data Adapter to Data Manager.

   For information about importing a Data Adapter, see *9.0 Importing a Data Adapter into Data Manager*.

4.1 Developing a Data Adapter using multiple iterations

It is a good practice to start a new Data Adapter XML file using the example file as a template. You should also create the Data Adapter in multiple iterations rather than attempting to complete the entire Data Adapter at one time.

1. In the first iteration, develop the XML file with header elements, server list query and a single map element.
2. Import the Data Adapter XML file into Data Manager.

   For more information, see *4.0 Best practices*.
3. The import process validates the Data Adapter and displays any errors and warnings.

   If your Data Adapter contains any errors, correct the XML and import it again until it validates successfully.
4. Create a data source using the new Data Adapter and load and migrate test data into Data Manager. You should also create some basic reports to ensure that the data loaded properly.
5. Continue to add elements to your Data Adapter in increments, repeating the testing process after each addition. In this way, it is easy to determine which element is causing an error.

5.0 Developing a Data Adapter for a CSV file source

This section describes the key elements required to develop a Data Adapter for data stored in CSV files. It includes an example showing the steps you can take to develop a DA to collect server metrics.

5.1 Before you begin

Before developing a CSV data source file, you should

- familiarize yourself with the structure of the CSV data source file.

  You should understand the data reported by the data source, units of the metrics, what the header column names mean, and so on. For additional information, see *5.2 CSV structure requirements and mapping*

- familiarize yourself with development best practices which include developing a Data Adapter in multiple iterations and starting with a template .xml file.

  For more information, see *4.0 Best practices*.

- review the template Data Adapter XML file provided in this kit (\examples\da_csv**generic_csv_template.xml**).

  This template Data Adapter contains mappings for all supported CSV data. You can use it as a template when developing your own custom Data Adapter for CSV data.

- review the example custom Data Adapter files included in this kit.

  The **server_config_and_disk_da.xml** supports gathering physical server configuration and disk data from generic CSV files.

  The **virtual_server_config_and_disk_da** supports gathering virtual server configuration and disk data from generic CSV files.

  For an example showing how to develop a custom Data Adapter, see *5.4 Developing a sample CSV Data Adapter*.

5.2 CSV structure requirements and mapping

CSV data files must meet the following structure requirements:

- Files must be well formatted and must have a header row for the names of the columns.
- Columns must be separated by commas.
- A single file must contain only the data for a single metric group/XML destination. For example, a single file can not contain metrics for both CPU and Disk.

5.3 CSV sample Data Adapter and associated CSV data files

The template Data Adapter for CSV data (**generic_csv_template.xml**) is mapped to columns within the sample data files included in the kit in the **examples\da_csv\sample_data** folder.

The following table shows the map ID and XML destination for each of the sample files.

Table 1: Mapping for Data Adapter for CSV files

Sample File	XML Map ID	XML Destination
GenericApplication.csv	applicationMetricsMap	application_metrics
GenericApplication_virtual.csv	virtualapplicationMetricsMap	application_metrics
GenericConfiguration.csv	serverMap	server
GenericConfiguration_virtual.csv	virtualserverMap	server_virtual
GenericCPU.csv	cpuMetricsMap	cpu_metrics
GenericCPU_virtual.csv	virtualcpuMetricsMap	cpu_metrics
GenericDisk.csv	diskMetricsMap	disk_metrics
GenericDisk_virtual.csv	virtualdiskMetricsMap	disk_metrics
GenericNIC.csv	networkMetricsMap	network_metrics
GenericNIC_virtual.csv	virtualnetworkMetricsMap	network_metrics
GenericProcess.csv	processMetricsMap	process_metrics
GenericProcess_virtual.csv	virtualprocessMetricsMap	process_metrics
GenericSystem.csv	serverMetricsMap	server_metrics
GenericSystem_virtual.csv	virtualserverMetricsMap	server_metrics
Generic_Cluster_Configuration.csv	serverMap_cluster	cluster
Generic_Cluster_Metrics.csv	clusterResourceMetricsMap	cluster_metrics
Generic_Resourcepool_Metrics.csv	PoolResourceMetricsMap	resource_pool_metrics
Generic_respool_configuration.csv	resourcepool_Map	resource_pool
Generic_transaction.csv	workloadTransactionMetricsMap	work_trans_metrics
Generic_workload.csv	workloadMetricsMap	workload_log

Note: Some of the column names in a few sample files include column headings with one or more space characters. The template Data Adapter illustrates how to use an alias to handle these columns. For more details, see *3.3.4.2 Space and special characters in source column name*.

5.4 Developing a sample CSV Data Adapter

The exercise in this section walks you through the steps to develop a Data Adapter for physical server configuration and disk data using the following sample files (included in the Custom Data Adapter SDK):

- **generic_csv_template.xml** - template Data Adapter XML file with which you will begin.
- **GenericConfiguration.csv** - sample CSV data file used as the source data for the server configuration data in the exercise.
- **GenericDisk.csv** - sample CSV data file used as the source data for the disk metrics in the exercise.
- **server_config_and_disk_da.xml** - the finished Data Adapter. You can use this file to confirm that you completed the steps in this exercise correctly.

5.4.1 Step 1: Define the XML file and add a single map element (serverMap)

In this step, you create a new Data Adapter XML file, then copy and customize the header information from the template.

1. Using a text or XML editor, copy the XML header information from the template file (**generic_csv_template.xml**) into your new file.
2. Modify the following XML elements in the header, as required:
   - Name
   - Author
   - Copyright
   - Description

   For information about these elements, see *3.2 XML elements*.

   Important: Data Manager requires a unique name for each Data Adapter that you import, and you cannot rename or reuse the name of any Data Adapter, even those that you are only testing during development. Therefore, you should use a temporary name for each development iteration. For example, use a name such as "*actual_name*_Iteration1". When you are satisfied with the Data Adapter, change the name to what you want it to be in production.

3. Locate and copy the "serverMap" element from the template file (**generic_csv_template.xml**) into your new file, placing it below the header information.

   Be sure to copy the entire element from <map id="serverMap"> to the final line </map>

4. At the end of the file, add a line as shown below to make the XML file syntactically correct.

   ```
   </data_source_type>
   ```

5.4.2 Step 2: Modify values for source section

For the exercise, you do not need to make any modification to the map file. However, for your own custom Data Adapter using a different source CSV file, you will want to modify the following items:

1. Replace the names of columns in the <source> element with the column names in your source CSV file. If a column name has space or special characters, use the alias feature to represent the column name.

   For more information, see *3.3.4.2 Space and special characters in source column name*.

   > **Tip**: You do not need to include all the columns in the source file. Include only those columns in which you are interested and remove the rest.

2. Change the data types of the columns as required. If the data type is “date” then you must specify the format of the date string as well. For a list of the allowed data types, see *3.3.4.3 Allowed data types for column elements*.

3. If a column in your source CSV file is not always present, set the **optional** flag to **true** for that column.

   For more details refer to section *3.3.4 Column elements*.

5.4.3 Step 3: Modify values for destination section

For the exercise, you do not need to make any modification to the map file. However, for your own custom Data Adapter using a different source CSV file, you will want to modify the name, units, and source information for all metrics.

1. In the source specification, you can specify the name of a column under <source> element or you can specify an arithmetic expression. For more information, see *3.3.5 Destinations element*.
2. Ensure that standard metrics units match those shown in the subsections in *10.0 XML destination details*. If the source data is in different units, you must create an arithmetic transformation.

   You can see a sample transformation in the example file (**generic_csv_ template.xml**). For information about standard metrics, units and so on for each XML destination, see *10.0 XML destination details*.
3. (*optional*) You can specify your own metrics and their units. Provide a valid description for these metrics using the <description> attribute of the <metric> element.

5.4.4 Step 4: Save the Data Adapter XML file

Save the file with an .xml extension.

Important: Be sure to save the initial iterations of the XML file using a temporary name. As with the XML element <name>, Data Manager requires that each Data Adapter XML file name be unique and will not allow you to change or reuse the file name later. For the preliminary iterations, use a name such as "*actual_name_* Iteration1.xml". When you complete the Data Adapter to your satisfaction, you can save the final file with the name you want.

5.4.5 Step 5: Import and test the Data Adapter

Before adding another map, be sure to import and test your Data Adapter. Doing so makes it much easier to locate and fix any errors.

1. Import the Data Adapter XML file into Data Manager.

   The import process validates the Data Adapter and displays any errors and warnings. For more information, see *9.0 Importing a Data Adapter into Data Manager*.

2. If there are any errors, correct the XML and import the Data Adapter again.
3. Create a data source using the new Data Adapter type, then use it to load and migrate data into Data Manager. You should also create some basic reports to verify that the data is loaded properly.

   For more information, see the *Data Manager Administration Guide*.

5.4.6 Step 6: Add a second map element (diskMetricsMap)

After verifying that your Data Adapter is working, add another map to the file.

1. Change the <name> element in the XML header (for example, *actual_name_* Iteration2.)
2. Locate and copy the "diskMetricsMap" element from the template file (**generic_csv_template.xml**) into your new file, placing it below the serverMap.

   Be sure to copy the entire element from <map id="diskMetricsMap"> to the final line </map>

3. Repeat steps 2 through 3 to modify the new map element.
4. Save the Data Adapter with a new file name (for example *actual_name_* Iteration2.xml).
5. Repeat Step 5 above to import and test the Data Adapter. If there are errors, you can focus on the new map element.

5.4.7 Step 7: Final steps

After completing the Data Adapter, ensure that the following are complete before you deploy it into the production environment:

1. Ensure that the description element includes all of the information about the Data Adapter that is important for users to know.
2. Ensure that the name of the Data Adapter (in the <name> element) is what you want for the final name.

   It is a good practice to have some version information as part of the name.
3. Ensure that the name of the Data Adapter XML file is what you want for the final name.

6.0 Developing a Data Adapter using logical grouping

This section describes the key elements required to develop a Data Adapter for data stored in CSV files which use the logical grouping feature of Data Manager. It includes an example showing the steps you can take to develop a DA to collect information by logical groups.

6.1 Before you begin

Before developing a CSV data source file using logical grouping, you should

- familiarize yourself with the structure requirements of the CSV data source file.

  The **GroupByCountry.csv** shows an example of a working CSV data source file with data supporting logical grouping.

- review the example custom Data Adapter files included in this kit.

  The **group_by_country_da.xml** supports gathering logical grouping data.

6.2 CSV sample Data Adapter and associated CSV data files

The template Data Adapter for CSV data (**generic_csv_template.xml**) is mapped to columns within the sample data files included in the kit in the **examples\da_csv\sample_data** folder.

The following table shows the map ID and XML destination for each of the sample files.

Table 1: Mapping for Data Adapter for CSV files

Sample File	XML Map ID	XML Destination
GroupByBusinessService.csv	GroupByBusinessService	group_config
GroupByCountry.csv	GroupByCountry	group_config
GroupByDatacenter.csv	GroupByDatacenter	group_config

6.3 Developing a sample CSV Data Adapter for logical groupings

The exercise in this section walks you through the steps to develop a Data Adapter for physical server configuration and disk data using the following sample files (included in the Custom Data Adapter SDK):

- **generic_csv_template.xml** - template Data Adapter XML file with which you will begin.
- **GroupByCountry.csv** - sample CSV data file used as the source data for the exercise.
- **group_by_country_da.xml** - the finished Data Adapter. You can use this file to confirm that you completed the steps in this exercise correctly.

6.3.1 Step 1: Define the XML file and add a single map element (GroupbyCountry)

In this step, you create a new Data Adapter XML file, then copy and customize the header information from the template.

1. Using a text or XML editor, copy the XML header information from the template file (**generic_csv_template.xml**) into your new file.
2. Modify the following XML elements in the header, as required:
   - Name
   - Author
   - Copyright
   - Description

   For information about these elements, see *3.2 XML elements*.

   Important: Data Manager requires a unique name for each Data Adapter that you import, and you cannot rename or reuse the name of any Data Adapter, even those that you are only testing during development. Therefore, you should use a temporary name for each development iteration. For example, use a name such as "*actual_name*_Iteration1". When you are satisfied with the Data Adapter, change the name to what you want it to be in production.

3. Locate and copy the "GroupByCountry" element from the template file (**generic_csv_template.xml**) into your new file, placing it below the header information.

   Be sure to copy the entire element from <map id="GroupByCountry"> to the final line </map>

4. At the end of the file, add a line as shown below to make the XML file syntactically correct.

   ```
   </data_source_type>
   ```

6.3.2 Step 2: Modify values for source section

For the exercise, you do not need to make any modification to the map file. However, for your own custom Data Adapter using a different source CSV file, you will want to modify the following items:

1. Replace the names of columns in the <source> element with the column names in your source CSV file. If a column name has space or special characters, use the alias feature to represent the column name.

   For more information, see *3.3.4.2 Space and special characters in source column name*.

   > **Tip**: You do not need to include all the columns in the source file. Include only those columns in which you are interested and remove the rest.

2. If a column in your source CSV file is not always present, set the **optional** flag to **true** for that column.

   For more details refer to section *3.3.4 Column elements*.

6.3.3 Step 3: Modify values for destination section

For the exercise, you do not need to make any modification to the map file. However, for your own custom Data Adapter using a different source CSV file, you will want to modify the name and source information for all metrics.

6.3.4 Step 4: Save the Data Adapter XML file

Save the file with an .xml extension.

> **Important**: Be sure to save the initial iterations of the XML file using a temporary name. As with the XML element <name>, Data Manager requires that each Data Adapter XML file name be unique and will not allow you to change or reuse the file name later. For the preliminary iterations, use a name such as "*actual_name_*Iteration1.xml". When you complete the Data Adapter to your satisfaction, you can save the final file with the name you want.

6.3.5 Step 5: Import and test the Data Adapter

Before adding another map, be sure to import and test your Data Adapter. Doing so makes it much easier to locate and fix any errors.

1. Import the Data Adapter XML file into Data Manager.

   The import process validates the Data Adapter and displays any errors and warnings. For more information, see *9.0 Importing a Data Adapter into Data Manager*.

2. If there are any errors, correct the XML and import the Data Adapter again.
3. Create a data source using the new Data Adapter type, then use it to load and migrate data into Data Manager. You should also create some basic reports to verify that the data is loaded properly.

   For more information, see the *Data Manager Administration Guide*.

4. (*optional*) Although not required for the exercise, you can add additional logical group maps to your Data Adapter and repeat the testing and verification process.

6.3.6 Step 6: Final steps

After completing the Data Adapter, ensure that the following are complete before you deploy it into the production environment:

1. Ensure that the description element includes all of the information about the Data Adapter that is important for users to know.
2. Ensure that the name of the Data Adapter (in the <name> element) is what you want for the final name.

   It is a good practice to have some version information as part of the name.

3. Ensure that the name of the Data Adapter XML file is what you want for the final name.

7.0 Developing a Data Adapter for a database source

Data Manager supports Data Adapters written for the following databases:

- Oracle
- Microsoft SQL server
- IBM DB2

7.1 Before you begin

Before you develop a Data Adapter for a database source, you must

- familiarize yourself with development best practices which include developing a Data Adapter in multiple iterations and starting with a template .xml file.

  For more information, see *4.0 Best practices*.

- have a good understanding of the schema of the database.
- determine the best way to get the data out of the database.

  For example, you should know if views or synonyms are available or whether you must write the SQL queries directly against the database tables.

- determine if the data is stored in pivoted fashion in tables/views/synonyms or whether the SQL query needs to pivot the data.
- select a database tool to develop SQL queries to gather data.
- obtain the following database credentials:
  - database host name/IP address
  - port on which database is listening
  - database type (Oracle, MS SQL server, DB2 etc)
  - database name
  - database schema name
  - database user name (with access permissions to select the data from various database objects)
  - password for the user name.

Note: The file **db_template.xml** available in this kit is a good starting point for creating a Data Adapter for databases.

7.2 Develop SQL queries

Using a database tool, develop SQL queries to

- gather a list of servers from the source database
- gather data of interest (such as CPU, disk, process metrics, and so on)

7.2.1 SQL query to obtain list of servers

Data Manager users specify the list of servers/components for which they want to gather data when using Data Adapters in Data Manager. Data Manager makes use of the server list query in the Data Adapter XML file to gather the list of servers/components and display it to the user on the GUI for selection. To display the appropriate list, you must develope a SQL query that returns only a distinct list of servers/components.

For example:

```
select distinct server_name from server_config
```

7.2.2 SQL query to obtain data

Develop queries to gather data of interest. Consider the following when developing queries to gather data:

Supported data

Ensure the data you are attempting to gather is supported. If there is no direct match you can store the data in a diverse metrics table. Supported metric/data groups are listed in *2.0 Supported data source metrics*.

Single metric or data group per query

Ensure a query returns data for a single metric or data group. For example, a single SQL query should not return both per Disk data and per NIC data. Separate metrics need to be in separate queries.

Pivoted data

Queries must return data in a pivoted fashion. For example, if a query returns system-level CPU utilization and system level memory utilization, the query needs to return them in two separate columns (not in name-value pairs). If the data is stored in a name-value pair in the source, use the SQL CASE statement or equivalent to pivot the data.

Data filter based on servers and date

To gather data from a database source, the Data Manager user must specify a list of servers and start and end date/times. If the data in a query should be filtered by dates and servers, the SQL query must include the appropriate filters in the WHERE clause of the statement. Depending on how data is stored in the source database, you might need to have one or both filters in the query. For example, the data for performance metrics should be filtered both by servers and time. However, configuration data might not include date/time information, in which case the query should filter the data only by servers.

- If the query needs to filter data based on servers only, the SQL query only needs to have the server filter in the WHERE clause of the query. For example:

```
SELECT server_name,
   time_period,
   GBL_CPU_TOTAL_UTIL,
   GBL_MEM_ UTIL
FROM server_KERNEL,
   server_SYSTEM
WHERE id_key_ = id_key
AND server_name IN (DGE_SERVERS_LIST)
```

- If the query needs to filter data based on date/time information and not by server, the query only needs to have the time filter in the WHERE clause of the query. Typically, the date/time filter has two components: Start Date/Time and End Date/Time. For example:

```
SELECT server_name,
   time_period,
   GBL_CPU_TOTAL_UTIL,
   GBL_MEM_ UTIL
FROM server_KERNEL,
   server_SYSTEM
WHERE id_key_ = id_key
AND time_period BETWEEN DGE_FROM_DATETIME AND DGE_TO_DATETIME
```

- If the query needs to filter data based on both servers and date/time information, the query needs to have filters for both date/time and servers in the WHERE clause of the query. In this case, time-based filtering must occur *before* the server filtering in the WHERE clause due to the way Data Manager handles this information in the query. For example:

  Important: Because of a known problem in the DGE, you must specify the date/time macros *before* the DGE_SERVERS_LIST macro in the WHERE clause.

```
SELECT server_name,
   time_period,
   GBL_CPU_TOTAL_UTIL,
   GBL_MEM_UTIL
FROM server_KERNEL,
   server_SYSTEM
WHERE id_key_ = id_key
AND time_period BETWEEN DGE_FROM_DATETIME AND DGE_TO_DATETIME
AND server_name IN (DGE_SERVERS_LIST)
```

Filtering data based on a threshold

If you expect to collect large amounts of data (such as process or disk metrics from large servers), consider collecting data based on a threshold. You can add a filter to the where clause of the query. For example, you could collect process data only when the process CPU utilization is greater than 5% and collect disk data only when disk utilization is greater than 10%.

Fully qualify database table names

If the user name being used to access data from the database is not the owner of that data schema, then you must qualify the table name with schema name. It will be in the form “SchemaName.TableName”

As clause to alias the column name

Use care when adding an "as" clause to alias column names. When the Data Adapter is imported, Data Manager generates a report that maps the source columns to the Data Manager fields/metrics. If you use an “as” clause to create an alias for a column name, the report cannot reflect the exact mappings to the source.

You can use the “as” clause to create aliases for column names in the following scenarios:

- If the column name has space or special characters.
- You are doing some kind of transformation either arithmetic or string on the data.

For example

```
Select (perc_usr + perc_sys) as cpu_utilization from cpu_table
```

Be sure to test the query in the SQL tool to make sure it is working as you intended.

Note: It is a good practice to develop a query for a single metric/data group, create the Data Adapter with that query only, and test the Data Adapter. When you are sure the Data Adapter is working properly, you can develop and add the rest of the required queries.

7.2.3 Modify the query to use Data Adapter macros

When you develop the SQL query to filter data based on servers and date/time, you should test it using real server names and dates. Be aware that you must replace absolute values for servers and dates with Data Adapter macros. For more information about macros, see *3.4 Data Adapter macros*.

- Start Date/time should be replaced with the macro DGE_FROM_DATETIME .
- End Date/time should be replaced with the macro DGE_TO_DATETIME .
- List of server names should be replaced with the macro DGE_SERVERS_LIST.

Data Manager substitutes user-supplied values (entered in the user interface) or these macros at run time while gathering data.

The following example query includes macros:

```
SELECT server_name,
  time_period,
  gbl_cpu_total_util,
  gbl_mem_util
FROM server_kernel,
  server_system
WHERE id_key_ = id_key
AND time_period BETWEEN DGE_FROM_DATETIME AND DGE_TO_DATETIME
AND server_name IN (DGE_SERVERS_LIST)
```

7.3 Developing a sample database Data Adapter

The exercise in this section walks you through the steps to develop a Data Adapter for physical server configuration and metrics using the following example files (included in the Custom Data Adapter SDK). The Data Adapter that you create will obtain data from a Data Manager 4.3 database.

- **db_template.xml** - template Data Adapter XML file with which you will begin.
- **hyperformix_datamanager_43_da.xml** - the finished Data Adapter. You can use this file to confirm that you completed the steps in this exercise correctly.

7.3.1 Step 1: Define the XML file and add a single map element (serverMap)

In this step, you create a new Data Adapter XML file, then copy and customize the header information from the template.

1. Using a text or XML editor, copy the XML header information from the template file (**db_template.xml**) into your new file.
2. Modify the following XML elements in the header, as required:
   - Name
   - Author
   - Copyright
   - Description

   For information about these elements, see *3.2 XML elements*.

   Important: Data Manager requires a unique name for each Data Adapter that you import, and you cannot rename or reuse the name of any Data Adapter, even those that you are only testing during development. Therefore, you should use a temporary name for each development iteration. For example, use a name such as "*actual_name*_Iteration1". When you are satisfied with the Data Adapter, change the name to what you want it to be in production.
3. Locate and copy the **serverMap** map element from the template file (**db_template.xml**) into your new file, placing it below the header information.

   Be sure to copy the entire element from <map id="serverMap"> to the final line </map>
4. At the end of the file, add a line as shown below to make the XML file syntactically correct.

   ```
   </data_source_type>
   ```

7.3.2 Step 2: Modify the value for the server list query

Change the SQL query for the server_list_query element with the query you developed for getting the list of servers.

7.3.3 Step 3: Modify map elements

Modify the source and destination map elements.

Source Section

1. Replace the select query in the select element with the query developed to gather system level metric information after replacing the server and date-time values with the Data Adapter macros. Refer to earlier sections for details.
2. Change the names of columns in the <source> element with the column names returned by the SQL statement. For more details, see *3.3.4 Column elements*.
3. Change the data types of the columns accordingly. For all the allowed data types see *3.3.4.3 Allowed data types for column elements*.

Destination Section

1. Change the source for the <required_fields> with the column names under the source element. For more information, see *3.3.5 Destinations element*.
2. If the GMT offset is not available in the source use the Data Adapter macro "DGE_GMT_OFFSET" for the required field "gmt_offset" as in the example DA.
3. Change the source for the <optional_fields>. If no data for the optional fields is present in the source data you need to remove <optional_fields> section.
4. Modify the name, units, and source information for metrics.
    a. In the source specification you can specify the name of a column in the <source> element or specify an arithmetic expression. For more information, see *3.3.5 Destinations element*.
    b. For information about standard metrics, units and so on for each XML destination, see *3.2 XML elements*.
    c. Standard metrics units must match those shown in *3.2 XML elements*. If the source data is in different units, you must create an arithmetic transformation.
    d. You can also specify your own metrics and their units. Provide a valid description for these metrics using the <description> attribute of the <metric> element.

7.3.4 Step 4: Save the Data Adapter XML file

Save the file with an .xml extension.

> **Tip**: Be sure to save the initial iterations of the XML file using a temporary name. Data Manager requires that each Data Adapter XML file name be unique and will not allow you to change the name later. For the preliminary iterations, use a name such as "Actual_Name_Iteration1.xml". When you complete the Data Adapter to your satisfaction, you can save the final file with the name you want.

7.3.5 Step 5: Import and test the Data Adapter

Import the Data Adapter XML file into Data Manager, validate the file, then load, test, and generate reports for the data. For more information, see *7.0 Developing a Data Adapter for a database source*.

7.3.6 Step 6: Add a second map element (serverMetricsMap)

After verifying that your Data Adapter is working, add another map to the file.

1. Change the <name> element in the XML header (for example, *actual_name_* Iteration2.)
2. Locate and copy the **serverMetricsMap** map element from the template file (**db_ template.xml**) into your new file, placing it below the serverMap.

   Be sure to copy the entire element from <map id="serverMetricsMap"> to the final line </map>
3. Repeat steps 2 through 3 to modify the new map element.
4. Save the Data Adapter with a new file name (for example *actual_name_* Iteration2.xml).
5. Repeat Step 5 above to import and test the Data Adapter. If there are errors, you can focus on the new map element.

7.3.7 Step 7: Final steps

After completing the Data Adapter, ensure that the following are complete before you deploy it into the production environment:

1. Ensure that the description element includes all of the information about the Data Adapter that is important for users to know.
2. Ensure that the name of the Data Adapter (in the <name> element) is what you want for the final name.

   It is a good practice to have some version information as part of the name.
3. Ensure that the name of the Data Adapter XML file is what you want for the final name.

8.0 Developing Data Adapters for diverse data and metrics

The method used to create a Data Adapter for diverse data or diverse metrics is very similar to that used to create a Data Adapter for CSV data. The Custom Data Adapter SDK provides sample Data Adapters for diverse data and diverse metrics and a number of files containing examples of source diverse data and diverse metrics. You can use these files as templates when creating your custom Data Adapters and source data files.

9.0 Importing a Data Adapter into Data Manager

After you develop a Data Adapter, import it into Data Manager, as follows:

1. Log on to Data Manager.
2. Click **Manage Source Types** in the navigation pane to display the Data Source Type Details page.
3. Click the **Add** button to display the Add Data Source Type screen.
4. Click **Browse** to select the .xml file you would like to import, then click **Open**.
5. Click **Validate** to make sure that the definition is valid and to display information about the type, such as the metrics it collects.

   If there are any errors, correct the Data Adapter XML and repeat the import and validation process until it is successfully imported.

   Note: The import process validates the Data Adapter and displays any errors and warnings. Adding a non-standard metric will always display a Warning message.

10.0 XML destination details

This topic includes details about the XML destination tables.

10.1 XML destination:application_metrics

Table 1: XML destination: application_metrics

XML destination	Metric group	Database staging table
application_metrics	Application level metrics	application_metrics_staging

10.1.1 Required fields

Table 2: Required fields

Element Name	Description
server_id	Name of the host/VM
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes
application_id	Application instance identifier. May be same as application _ name.
application_name	Application Name. May be same as application_id.

10.1.2 Optional fields

Table 3: Optional fields

Element Name	Description
duration	Duration of the metric in seconds.

10.1.3 Standard metrics

Table 4: Standard metrics

Metric Name	Metric Description	Metric Unit	Used by CM
APP_CPU_UTIL	CPU Utilization	percent	X
APP_DISK_PHYS_RDWR_RATE	Disk Input/Output Rate	transfers per second	
APP_DISK_PHYS_READ_RATE	Disk reads per second	transfers per second	
APP_DISK_PHYS_WRITE_RATE	Disk writes per second	transfers per second	
APP_MEM_PRIVATE_KILOBYTES	Private Memory	kilobytes	
APP_MEM_UTIL	Memory Utilization	percent	
APP_PHYS_CPU_UTIL	Physical CPU Utilization	percent	
APP_PHYS_MEM_UTIL	Physical Memory Utilization	percent	
APP_VIRT_CPU_UTIL	Virtual CPU Utilization	percent	X
APP_VIRT_MEM_UTIL	Virtual Memory Utilization	percent	

10.2 XML destination : cluster

Table 1: XML destination: cluster

XML destination	Metric group	Database staging table
cluster	Configuration data for the cluster.	cluster_staging

10.2.1 Required fields

Table 2: Required fields

Element Name	Description
cluster_id	Cluster Identifier. Could be Name cluster as well.
cluster_name	Name of the cluster.
src_start_date	Time at which configuration data was reported

10.2.2 Optional fields

Table 3: Optional fields

Element Name	Description
appl_server_motion_count	In clusters such as VMware ESX clusters this field indicates how many times VMotion happened within the cluster. Use this field to indicate similar statistics for other clusters.
failover_level	What is the fail over level of the cluster? In a High availability clusters this is a reflection of maximum number of hosts within a cluster that can fail simultaneously and still expect the cluster to continue functioning.
ha_enabled	Flag to indicate if the cluster is enabled for high availability Allowed values are 0 – Disabled 1 – Enabled
load_balancing_enabled	Flag to indicate if the cluster is enabled for load balancing. Allowed values are 0 – Disabled 1 – Enabled
power_mgmt_enabled	Flag to indicate if the cluster is enabled for advanced power management. For example power off a host in a cluster if load on the cluster is very light. Allowed values are 0 – Disabled 1 – Enabled
power_off_on_isolation	Flag to indicate if power off on isolation feature is enabled. Allowed values are 0 – Disabled 1 – Enabled

10.2.3 Standard metrics

No metrics is stored for this group. Only configuration data is stored.

10.3 XML destination : cluster_metrics

Table 1: XML destination: cluster_metrics

XML destination	Metric group	Database staging table
cluster_metrics	Cluster metrics.	cluster_metrics_staging

10.3.1 Required fields

Table 2: Required fields

Element Name	Description
cluster_id	Cluster Identifier. Could be Name cluster as well.
cluster_name	Name of the cluster.
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes

10.3.2 Optional fields

Table 3: Optional fields

Element Name	Description
duration	Duration of the metric in seconds.

10.3.3 Standard metrics

Table 4: Standard metrics

Metric Name	Metric Description	Metric Unit
CLUSTER_CPU_CORE_USAGE	Total CPU cores used by the cluster. This is the sum of CPU cores used by all active hosts in the cluster	total number of cores
CLUSTER_CPU_MHZ_USAGE	Total CPU MHz used by the cluster. The sum of MHz used by all active hosts in the cluster.	megahertz
CLUSTER_CPU_THREAD_USAGE	Total CPU threads used by the cluster. This is the sum of CPU threads used by all active hosts in the cluster	total number of threads
CLUSTER_CPU_UTIL	Percent CPU utilization by the cluster	percent
CLUSTER_EFFECTIVE_CPU_MHZ	Effective CPU in MHz that is available to the cluster.	megahertz
CLUSTER_EFFECTIVE_MEM_MB	Effective memory in Mega bytes that is available to the cluster.	megabytes
CLUSTER_FAILOVER_LEVEL	Failover level of the cluster.	numeric flag
CLUSTER_MEM_MB_USAGE	Total Memory in Megabytes used by the cluster. The sum of memory used by all active hosts in the cluster.	megabytes
CLUSTER_MEM_UTIL	Percent memory utilization by the cluster	percent

10.4 XML destination : cpu_metrics

Table 1: XML destination: cpu_metrics

XML destination	Metric group	Database staging table
cpu_metrics	Per processor metrics	cpu_metrics_staging

10.4.1 Required fields

Table 2: Required fields

Element Name	Description
server_id	Name of the host/VM
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes
cpu_id	CPU instance identifier
cpu_name	CPU instance identifier name. May be same as cpu_id

10.4.2 Optional fields

Table 3: Optional fields

Element Name	Description
duration	Duration of the metric in seconds.

10.4.3 Standard metrics

Table 4: Standard metrics

Metric Name	Metric Description	Metric Unit	Used by CM
CPU_CPU_UTIL	CPU Utilization	percent	X
CPU_LS_PHYS_CPU_UTIL	Per physical CPU Utilization for logical server	percent	
CPU_PROC_QUE_LEN	Average Length of the Queue	processes	
CPU_WAIT_TIME	CPU Wait Time	seconds	

10.5 XML destination : disk_metrics

Table 1: XML destination: disk_metrics

XML destination	Metric group	Database staging table
disk_metrics	Per physical disk metrics	disk_metrics_staging

10.5.1 Required fields

Table 2: Required fields

Element Name	Description
server_id	Name of the host/VM
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes
disk_id	Physical Disk instance identifier
disk_name	Physical Disk identifier. May be same as disk_id

10.5.2 Optional fields

Table 3: Optional fields

Element Name	Description
duration	Duration of the metric in seconds.

10.5.3 Standard metrics

Table 4: Standard metrics

Metric name	Metric description	Metric unit	Used by CM
DISK_AVG_QUE_LEN	Average Length of the Queue	transfers	
DISK_AVG_SERVICE_TIME	Average Service Time	seconds	
DISK_AVG_WAIT_TIME	Average Wait Time	seconds	
DISK_LS_READ_BYTE_RATE	Logical System Read Byte Rate	bytes per second	
DISK_LS_TOTAL_BYTE_RATE	Logical System Total Byte Rate	bytes per second	
DISK_LS_UTILIZATION	Logical Server Busy Utilization	percent	X
DISK_LS_WRITE_BYTE_RATE	Logical System Write Byte Rate	bytes per second	
DISK_PHYS_RDWR_RATE	Physical Read/Write Rate	transfers per second	
DISK_PHYS_READS	Total Physical Reads	transfers	
DISK_PHYS_READ_RATE	Rate of Physical Reads	transfers per second	
DISK_PHYS_WRITES	Total Physical Writes	transfers	
DISK_PHYS_WRITE_RATE	Rate of Physical Writes	transfers per second	
DISK_READ_BYTE_RATE	Read Byte Rate	bytes per second	
DISK_READ_SERVICE_TIME	Read Service Time	seconds	
DISK_TOTAL_BYTE_RATE	Total Byte Rate	bytes per second	
DISK_TOTAL_IO	Total number of IOs done	transfers	
DISK_UTILIZATION	Busy Utilization	percent	X
DISK_WRITE_BYTE_RATE	Write Byte Rate	bytes per second	
DISK_WRITE_SERVICE_TIME	Write Service Time	seconds	

10.6 XML destination : diverse_data

Table 1: XML destination: diverse_data

XML destination	Metric group	Database staging table
diverse_data	Diverse Data	diverse_data_staging

10.6.1 Required fields

Table 2: Required fields

Element name	Description
category	Category that the data belongs to.
gmt_offset	GMT offset in minutes

10.6.2 Optional fields

Table 3: Optional fields

Element name	Description
subcategory	Subcategory that the data belongs to. Could be used as one of the keys to identity the data.
grouping	Group that the data belongs to. Could be used as one of the keys to identity the data.
class	Class that the data belongs to. Could be used as one of the keys to identity the data.
source	Source of data.
attribute1	An attribute of the data. Could be used as one of the keys to identity the data.
attribute2	An attribute of the data. Could be used as one of the keys to identity the data.
attribute3	An attribute of the data. Could be used as one of the keys to identity the data.
attribute4	An attribute of the data. Could be used as one of the keys to identity the data.
attribute5	An attribute of the data. Could be used as one of the keys to identity the data.
std_metric_name	Metric name if the data represented is a metric.
metric_description	Description of the metric.
metric_time	Time at which metric was reported
metric_value	Numeric value of the metric if any.
metric_value_char	Character value of the metric if any.
duration	Duration of the metric in seconds.

10.6.3 Standard metrics

No standard metrics is stored for this group.

10.7 XML destination : diverse_metrics

Table 1: XML destination: diverse_metrics

XML destination	Metric group	Database staging table
diverse_ metrics	Diverse metrics	diverse_metrics _staging

10.7.1 Required fields

Table 2: Required fields

Element name	Description
category	Category that the data belongs to.
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes

10.7.2 Optional fields

Table 3: Optional fields

Element name	Description
subcategory	Subcategory that the data belongs to. Could be used as one of the keys to identity the data.
grouping	Group that the data belongs to. Could be used as one of the keys to identity the data.
class	Class that the data belongs to. Could be used as one of the keys to identity the data.
source	Source of data.
attribute1	An attribute of the data. Could be used as one of the keys to identity the data.
attribute2	An attribute of the data. Could be used as one of the keys to identity the data.
attribute3	An attribute of the data. Could be used as one of the keys to identity the data.
attribute4	An attribute of the data. Could be used as one of the keys to identity the data.
attribute5	An attribute of the data. Could be used as one of the keys to identity the data.
duration	Duration of the metric in seconds.

10.7.3 Standard metrics

No standard metrics is stored for this group. Users can define own metrics.

10.8 XML destination : group_config

Table 1: XML destination: group_config

XML destination	Metric group	Database staging table
group_config	Group configuration data	group_config_staging

10.8.1 Required fields

Table 2: Required fields

Element Name	Description
entity_name	Name of the entity being grouped (for example, server name, VM name, workload name, and so on).
entity_type	Type of entity being grouped (for example, Server, VM, Workload, Transaction, and so on).
src_start_date	Time at which the entity grouping was reported.
gmt_offset	GMT offset in minutes.

10.8.2 Optional fields

Table 3: Optional fields

Element Name	Description
status	Status of the grouping (for example, active or inactive).
end_date	Date at which the entity is no longer in the group.
effective_date	Date at which the entity becomes a member of the group.

10.8.3 Group fields

Table 4: Group fields

Element Name	Description
group	Name of one or more groups to which the entity is assigned. Groups can be nested to create a hierarchy. An entity can belong to more than one group.

10.8.4 Standard metrics

No metrics are stored for this group. Only group configuration data is stored.

10.9 XML destination : network_metrics

Table 1: XML destination: diverse_metrics

XML destination	Metric group	Database staging table
network_metrics	Per network interface metrics	network_metrics_staging

10.9.1 Required fields

Table 2: Required fields10.9 XML destination : network_metrics

Element name	Description
server_id	Name of the host/VM
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes
interface_id	Network Interface instance identifier
interface_name	Network Interface identifier. May be same as interface _id

10.9.2 Optional fields

Table 3: Optional fields

Element name	Description
duration	Duration of the metric in seconds.

10.9.3 Standard metrics

Table 4: Standard metrics

Metric name	Metric description	Metric unit	Used by CM
NETWORK_BYTES_IN	Bytes Input	bytes	X
NETWORK_BYTES_OUT	Bytes Out	bytes	X
NETWORK_BYTES_RECV_RATE	Total number of bytes per second received by the interface	bytes per second	X
NETWORK_BYTES_SEND_RATE	Total number of bytes per second sent by the interface	bytes per second	X
NETWORK_BYTES_TOTAL	Total Bytes	number of bytes	
NETWORK_BYTES_TOTAL_RATE	Total number of bytes per second sent and received by the interface	bytes per second	X
NETWORK_LS_BYTES_RECV_RATE	Total number of bytes per second received by the logical server interface	bytes per second	X
NETWORK_LS_BYTES_SEND_RATE	Total number of bytes per second sent by the logical server interface	bytes per second	X
NETWORK_LS_BYTES_TOTAL_RATE	Logical system total number of bytes per second sent and received by the interface	bytes per second	
NETWORK_TOTAL_BANDWIDTH	Bandwidth	bits per second	X

10.10 XML destination : process_metrics

Table 1: XML destination: process metrics

XML destination	Metric group	Database staging table
process_metrics	Process level metrics	process_metrics_staging

10.10.1 Required fields

Table 2: Required fields

Element name	Description
server_id	Name of the host/VM
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes
process_id	Process instance identifier. May be same as process_name.
process_name	Process Name. May be same as process_id.

10.10.2 Optional fields

Table 3: Optional fields

Element name	Description
command_line	Command Line arguments to the process
duration	Duration of the metric in seconds.
user_name	User name that is executing the process

10.10.3 Standard metrics

Table 4: Standard metrics

Metric name	Metric description	Metric unit	Used by CM
PROC_CPU_UTIL	CPU Utilization	percent	X
PROC_CPU_UTIL_NORMALIZATION	CPU Utilization Normalization	percent	
PROC_DISK_PHYS_RDWR_RATE	Disk Input/Output Rate	transfers per second	
PROC_DISK_PHYS_READ_RATE	PROC_DISK_PHYS_READ_ RATE	transfers per second	
PROC_DISK_PHYS_WRITE_RATE	PROC_DISK_PHYS_WRITE_ RATE	transfers per second	
PROC_DISK_READ_BYTE_RATE	Disk Read Byte Rate	bytes per second	
PROC_DISK_TOTAL_BYTE_RATE	PROC_DISK_TOTAL_BYTE_ RATE	bytes per second	
PROC_DISK_TOTAL_IO	Total Input/Output	number	
PROC_DISK_UTILIZATION	Disk Utilization by Process	percent	
PROC_DISK_WRITE_BYTE_RATE	Disk Write Byte Rate	bytes per second	
PROC_IO_PHYS_RDWR_RATE	Disk Input/Output Rate	transfers per second	
PROC_IO_PHYS_READ_RATE	PROC_IO_PHYS_READ_ RATE	transfers per second	
PROC_IO_PHYS_WRITE_RATE	PROC_IO_PHYS_WRITE_ RATE	transfers per second	
PROC_IO_READ_BYTE_RATE	PROC_IO_READ_BYTE_ RATE	bytes per second	
PROC_IO_TOTAL_BYTE_RATE	PROC_IO_TOTAL_BYTE_ RATE	bytes per second	
PROC_IO_WRITE_BYTE_RATE	PROC_IO_WRITE_BYTE_ RATE	bytes per second	
PROC_MEM_PRIVATE_KILOBYTES	Private Kilobytes	kilobytes	

Metric name	Metric description	Metric unit	Used by CM
PROC_MEM_UTIL	Memory Utilization	percent	
PROC_MEM_VIRTUAL_KILOBYTES	Virtual Kilobytes	kilobytes	
PROC_NETWORK_UTILIZATION	Network Utilization	percent	
PROC_PHYS_CPU_UTIL	Physical CPU Utilization	percent	
PROC_PHYS_MEM_UTIL	Physical Memory Utilization	percent	
PROC_VIRT_CPU_UTIL	Virtual CPU Utilization	percent	X
PROC_VIRT_MEM_UTIL	Virtual Memory Utilization	percent	

10.11 XML destination : resource_pool

Table 1: XML destination: resource_pool

XML destination	Metric group	Database staging table
resource_pool	Configuration data for the resource pool.	resource_pool_staging

10.11.1 Required fields

Table 2: Required fields

Element Name	Description
pool_id	Resource pool Identifier. Could be Name resource pool as well.
pool_name	Name of the resource pool.
src_start_date	Time at which configuration data was reported

10.11.2 Optional fields

Table 3: Optional fields

Element Name	Description
cpu_allocated	Entitled CPU capacity in native units for the pool. For example in case of AIX this is number of CPU cores entitled.
cpu_expandable_flag	Indicates if the pool is enabled to borrow CPU resources from its parent if needed if the virtualization technology supports this feature. Allowed values are 0. Disabled 1. Enabled
cpu_max	Maximum CPU resources in native units allocated to the pool. In case of VMware this is maximum CPU MHz (Limit) and in case of AIX this is maximum number of CPU cores.
cpu_min	Minimum CPU resources in native units allocated to the pool. In case of VMware this would be minimum MHz (Reservation) allocated; In case of AIX this would be minimum number of CPU cores allocated.
cpu_shares	CPU shares in native units allocated to the pool.
mem_allocated	Entitled memory in Magabytes for the pool.
mem_expandable	Indicates if the pool is enabled to borrow memory resources from its parent if needed if the virtualization technology supports this feature. Allowed values are 0 - Disabled 1 - Enabled
mem_max	Maximum memory in Megabytes allocated to the pool.
mem_min	Minimum memory in Megabytes allocated to the pool.
mem_shares	Memory shares in native units allocated to the pool.
parent_cluster_name	Name of the parent cluster if the pool is part of a cluster
parent_host_name	Name of the physical host is the pool is part of the host.
parent_id	ID of the parent of the pool. Could be same as parent name.
parent_pool_name	Name of the pool name if a pool is a child of another resource pool.

Element Name	Description
parent_type	Type of the parent of the resource pool. Allowed values are 1 – Host 2 – Virtual Machine 3 - Resource Pool 4 - Cluster

Note: In general parent information such as parent_pool_name or parent_cluster_ name is optional. But if any of the parent information is specified then parent_id and parent_type must be specified as well. Note that parent_id can be same as the parent name. parent_id is the name/id of the immediete parent. For example if POOL_A is the child of POOL_B and they both are running in CLUSTER_C, then for POOL_A parent_pool_name will be POOL_B and parent_cluster_name will be CLUSTER_C. parent_id for POOL_A will be the id/name of POOL_B and parent_type will be 3(Resource Pool)

10.11.3 Standard metrics

No metrics is stored for this group. Only configuration data is stored.

10.12 XML destination : resource_pool_metrics

Table 1: XML destination: resource_pool_metrics

XML destination	Metric group	Database staging table
resource_pool_metrics	Resource pool metrics.	resource_pool_metrics _staging

10.12.1 Required fields

Table 2: Required fields

Element name	Description
pool_id	Resource pool Identifier. Could be Name resource pool as well.
pool_name	Name of the resource pool.
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes

10.12.2 Optional fields

Table 3: Optional fields

Element name	Description
duration	Duration of the metric in seconds

10.12.3 Standard metrics

Table 4: Standard metrics

Metric name	Metric description	Metric unit
RESPOOL_TOTAL_CPU_CORE_USAGE	Total CPU cores used by the Virtual Machines that are direct children of the resource pool and all child resource pools including all the children in the hierarchy.	number of cores
RESPOOL_TOTAL_CPU_MHZ_USAGE	Total CPU MHz used by the Virtual Machines that are direct children of the resource pool and all child resource pools including all the children in the hierarchy.	megahertz
RESPOOL_TOTAL_CPU_THREAD_USAGE	Total CPU threads used by the Virtual Machines that are direct children of the resource pool and all child resource pools including all the children in the hierarchy.	number of threads
RESPOOL_TOTAL_CPU_UTIL	Percent CPU utilization by the resource pool.	percent
RESPOOL_TOTAL_MEM_MB_USAGE	Total Memory in MB used by the Virtual Machines that are direct children of the resource pool and all child resource pools including all the children in the hierarchy.	megabytes
RESPOOL_TOTAL_MEM_UTIL	Percent memory utilization the resource pool.	percent
RESPOOL_VM_CPU_MHZ_USAGE	Total CPU MHz used by the Virtual Machines that are the direct children of the resource pool.	megahertz
RESPOOL_VM_MEM_MB_USAGE	Total Memory in MB used by the Virtual Machines that are the direct children of the resource pool.	megabytes

10.13 XML destination : server

Table 1: XML destination : server

XML destination	Metric group	Database staging table
server	Configuration data for the Host.	server_staging

10.13.1 Required fields

Table 2: Required fields10.13 XML destination : server

Element name	Description
server_id	Host Identifier. Could be Name of the host or IP address
src_start_date	Time at which configuration data was reported
host_name	Name of the host.

10.13.2 Optional fields

Table 3: Optional fields

Element name	Description
cores_per_processor_chip	Number of CPU cores per processor chip.
cpu_clock_rate	CPU clock rate in MHz
cpu_utilization_type	Type of CPU Utilization reported by the monitor. Monitoring agents could report traditional utilization, capacity utilization, purr utilization, core utilization etc. For example agents running on regular X86 servers report traditional utilization where as agents running on AIX 5 and 6 systems report PURR utilization. Based on this value Hyperformix Modeling tools such as Capacity Manager interprets the data differently. Allowed values are 0 – Unknown 1 - Regular(Dispatcher) Utilization 2 - Capacity Utilization 3 - Purr Utilization 4 - Core Utilization 5 - MHz utilization
datacenter_name	Name of the data center that the host belongs to.
hardware_family	Hardware family. For example, PowerEdge or BladeCenter.
hardware_vendor	The manufacturer of the hardware. For example, Dell or IBM.
live_migration_enabled	When a host is used as virtualization host to host virtual machines, it can be enabled/disabled to move running virtual machine across hosts provided the underlying virtualization technology supports it. For example VMware calls this VMotion; Citrix Xen calls this Xen motion etc. Use this flag to indicate if this feature is enabled on the host. Allowed values are 0 – Disabled 1- Enabled
num_cores	Number of CPU cores on the host
num_cpus	Number of CPU threads on the host.
num_disk	Number of physical disks on the host.

Element name	Description
num_network_interfaces	Number of network interfaces on the host.
num_processor_chips	Number of processor chips on the host.
os_type	Operating System running on the host.
parent_cluster_name	Name of the parent cluster if the host is part of a cluster
parent_id	ID of the parent if the host is part of a grouping such as cluster. Could be same as parent name.
parent_pool_name	Name of the pool name if the host is part of a grouping such as a resource pool.
parent_type	Type of the parent if the host is part of a grouping such as cluster. Allowed values are 1 – Host 3 - Resource Pool 4 - Cluster
processor_family	Processor family.
system_type	Type of the host machine such as "HP Proliant DL" etc.
threads_per_processor_core	Number of CPU threads per processor core.
tlp_enabled_flag	Flag to indicate whether thread level processing is enabled. Valid values are 0 - Disabled 1 - Enabled
total_memory	Total Physical Memory in kilo bytes.
virtualization_solution	Name of the Virtualization solution such as "VMware ESX 3.5" if the host is a virtualization host.

Note: In general parent information such as parent_pool_name or parent_cluster_ name is optional. But if any of the parent information is specified then parent_id and parent_type must be specified as well. Note that parent_id can be same as the parent name. parent_id is the name/id of the immediete parent. For example if HOST_A is running on CLUSTER_C , then for HOST_A parent_cluster_name will be CLUSTER_C. parent_id for HOST_A will be the id/name of CLUSTER_ C and parent_type will be 4(cluster)

10.13.3 Standard metrics

No metrics is stored for this group. Only configuration data is stored.

10.14 XML destination : server_metrics

Table 1: XML destination : server_metrics

XML destination	Metric group	Database staging table
server_metrics	System level metrics	server_metrics_staging

10.14.1 Required fields

Table 2: Required fields

Element name	Description
server_id	Name of the host/VM
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes

10.14.2 Optional fields

Table 3: Optional fields

Element name	Description
duration	Duration of the metric in seconds.

10.14.3 Standard metrics

Table 4: Standard metrics

Metric Name	Metric description	Metric unit	Used by CM
GBL_CPU_MHZ_USAGE	Total CPU MHz used by the host	megahertz	
GBL_CPU_PROC_QUE_LEN	Length of the Process Queue	processes	
GBL_CPU_UTIL	CPU Utilization	percent	X
GBL_DISK_AVG_QUE_LEN	Average Length of the Queue	transfers	
GBL_DISK_PHYS_RDWR_RATE	Physical Read/Write Rate	transfers per second	
GBL_DISK_PHYS_READS	Total Physical Reads	transfers	
GBL_DISK_PHYS_READ_RATE	Rate of Physical Reads	transfers per second	
GBL_DISK_PHYS_WRITES	Total Physical Writes	transfers	
GBL_DISK_PHYS_WRITE_RATE	Rate of Physical Writes	transfers per second	
GBL_DISK_READ_BYTE_RATE	Read Byte Rate	bytes per second	
GBL_DISK_READ_SERVICE_TIME	Read Service Time	seconds	
GBL_DISK_SERVICE_TIME	Average Service Time	seconds	
GBL_DISK_TOTAL_BYTE_RATE	Total Byte Rate	bytes per second	
GBL_DISK_TOTAL_IO	Total Input/Output	number	
GBL_DISK_UTILIZATION	Busy Utilization	percent	
GBL_DISK_WRITE_BYTE_RATE	Write Byte Rate	bytes per second	
GBL_DISK_WRITE_SERVICE_TIME	Write Service Time	seconds	
GBL_LS_DISK_TOTAL_BYTE_RATE	Logical System Total Byte Rate	bytes per second	
GBL_LS_MEM_MB_USAGE	Total Memory in Megabytes used by the VM	megabytes	
GBL_LS_NETWORK_BYTES_TOTAL_RATE	Total number of network bytes per second by the logical server	bytes per second	

Metric Name	Metric description	Metric unit	Used by CM
GBL_LS_PHYS_CPU_MHZ_USED	Total physical megahertz used by the logical system.	megahertz	
GBL_LS_PHYS_CPU_USED	Number of physical CPUs used by the logical system	number of cores	
GBL_LS_PHYS_CPU_UTIL	Logical system physical CPU Utilization	percent	X
GBL_LS_PHYS_MEM_USED	Logical System Memory Used	megabytes	
GBL_LS_PHYS_MEM_UTIL	Logical System physical memory utilization	percent	X
GBL_LS_POOL_CPU_IDLE	Pool CPU idle time the VM	percent	
GBL_LS_POOL_CPU_UTIL	Percentage of Pool CPU Utilized by the VM	percent	
GBL_LS_VIRT_CPU_UTIL	Logical System virtual CPU Utilization	percent	
GBL_LS_VIRT_MEM_UTIL	Logical System Virtual Memory Utilization	percent	
GBL_MEM_AVAILABLE_BYTES	Available Memory in Bytes	bytes	
GBL_MEM_COMMITTED_BYTES	Committed Memory in Bytes	bytes	
GBL_MEM_MB_USAGE	Total Memory in Megabytes used by the host	megabytes	
GBL_MEM_TOTAL	Total Memory	bytes	
GBL_MEM_USED	Memory Used	kilobytes	
GBL_MEM_UTIL	Memory Utilization	percent	X
GBL_NETWORK_BYTES_RECV_RATE	Total number of bytes per second received by the server	bytes per second	
GBL_NETWORK_BYTES_SEND_RATE	Total number of bytes per second sent by the server	bytes per second	
GBL_NETWORK_BYTES_TOTAL_RATE	Total number of bytes per second sent and received by the server	bytes per second	
GBL_RUN_QUEUE	Length of the Process Queue	processes	

10.15 XML destination : server_virtual

Table 1: XML destination : server_virtual

XML destination	Metric group	Database staging table
server_virtual	Configuration data for the virtual machines.	server_virtual_staging

10.15.1 Required fields

Table 2: Required fields

Element name	Description
cpu_resource_alloc_unit	In which units CPU resources are allocated to Virtual Machines. For example VMware allocates CPU resources in terms of Mhz per VM; IBM Power Systems allocates cpu cores per VM. Mention a string such as "MHz per VM" or "Cores per VM" etc for this field.
online_virt_cpu	Number of virtual cpus in the VM
partition_name	Name of the virtual machine. Please note that this may or may not be the DNS name of the OS running on the VM. Virtualization technologies assigns a name for a virtual partition when it is created before a OS is installed on that partition. If such name is not available use the DNS name of the OS running on the VM.
physical_host	Name of the physical host on which VM is running
server_id	DNS name of the virtual machine. Could be Name or IP address
src_start_date	Time at which configuration data was reported
virtualization_solution	Name of the Virtualization solution such as "VMware ESX 3.5" if the host is a virtualization host.

10.15.2 Optional fields

Table 3: Optional fields

Element name	Description
active_phys_cpu	Number of active CPU threads on the physical host.
cpu_utilization_type	Type of CPU Utilization reported by the monitor. Monitoring agents could report traditional utilization, capacity utilization, purr utilization, core utilization etc. For example agents running on regular X86 servers report traditional utilization where as agents running on AIX 5 and 6 systems report PURR utilization. Based on this value Hyperformix Modeling tools such as Capacity Manager interprets the data differently. Allowed values are 0 – Unknown 1 - Regular(Dispatcher) Utilization 2 - Capacity Utilization 3 - Purr Utilization 4 - Core Utilization 5 - MHz utilization
datacenter_name	Name of the data center that the host belongs to.
entitled_capacity	Entitled CPU capacity in native units for the VM. For example in case of AIX this is number of CPU cores entitled.
group_name	Name of the group this VM belongs to if there is any sort of grouping.
max_capacity	Maximum CPU resources in native units allocated to the VM. In case of VMware this is maximum CPU MHz (Limit) and in case of AIX this is maximum number of CPU cores.
max_memory	Maximum memory in Megabytes allocated to the VM.
max_phys_cpu	Maximum number of CPUs the physical host could have.
max_virt_cpu	Maximum number of virtual CPUs on the VM
memory_overhead	Memory overhead of the VM in bytes.
memory_share	Memory shares in native units allocated to the VM.
min_capacity	Minimum CPU resources in native units allocated to the VM. In case of VMware this would be minimum MHz (Reservation) allocated; In case of AIX this would be minimum number of CPU cores allocated.

Element name	Description
min_memory	Minimum memory in Megabytes allocated to the VM.
min_virt_cpu	Minimum number of virtual CPUs on the VM
num_logical_processors	Number of logical processors per VM.
online_memory	Entitled Memory in kilo bytes allocated to VM.
os_type	Operating System running on the VM.
parent_cluster_name	Name of the parent cluster if the host/VM is part of a cluster
parent_id	ID of the parent if the host/VM is part of a grouping such as cluster. Could be same as parent name.
parent_pool_name	Name of the pool name if the VM is part of a resource pool.
parent_type	Type of the parent if the host is part of a grouping such as cluster. Allowed values are 1 – Host 2 – Virtual Machine 3 - Resource Pool 4 - Cluster
partition_mode	Indicates if the VM is capped for its CPU and Memory resources or not. Allowed values are • capped • uncapped
partition_number	Number of the partition if any partition number is assigned to a VM by a virtualization technology.
partition_type	Indicates if the CPU and Memory resources are dedicated to a VM or shared. Allowed values are • shared • dedicated
threads_per_processor_core	Number of CPU threads per processor core
tlp_enabled_flag	Flag to indicate whether thread level processing is enabled. Valid values are 0 - Disabled 1 - Enabled
var_capacity_weight	CPU shares in native units allocated to the VM

Note: In general parent information such as parent_pool_name or parent_cluster_name is optional. But if any of the parent information is specified then parent_id and parent_type must be specified as well. Note that parent_id can be same as the parent name. parent_id is the name/id of the immediete parent. For example if VM_A is running on POOL_B and they both are running in CLUSTER_C, then for VM_A parent_pool_name will be POOL_B and parent_cluster_name will be CLUSTER_C. parent_id for VM_A will be the id/name of POOL_B and parent_type will be 3(Resource Pool)

10.15.3 Standard metrics

No metrics is stored for this group. Only configuration data is stored.

10.16 XML destination : workload_log

Table 1: XML destination : workload_log

XML destination	Metric group	Database staging table
workload_log	Workload metrics	workload_log_staging

10.16.1 Required fields

Table 2: Required fields

Element name	Description
workload_type	Type of workload such as IT service or Business service etc.
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes

10.16.2 Optional fields

Table 3: Optional fields

Element name	Description
application_name	Name of the Business Process. A Business process may have zero, one or more business services or IT Services.
business_function_name	Name of the IT service. A IT Service may have zero, one or more Sub Systems (Applications) or Business Services. Applications run a host and application metrics can be imported to DM using standard application_metrics_staging table.
duration	Duration of the metric in seconds.
workload_name	Name of the Business Service. A Business Service may have zero , one or more IT services or Business Processes

10.16.3 Standard metrics

Table 4: Standard metrics

Metric name	Metric description	Metric unit
WKLOAD_ARRIVAL_COUNT	Number of units of work that arrived in the measurement interval.	units of work
WKLOAD_COMPLETION_COUNT	Number of units of work that completed in the measurement interval.	units of work
WKLOAD_CURRENT_EXECUTIONS	Number of work units in progress at the end of the measurement interval.	units of work
WKLOAD_RESPONSETIME_AVERAGE	Average response time in seconds of units of work that completed.	seconds
WKLOAD_RESPONSETIME_MAXIMUM	Maximum response time in seconds of completed work units.	seconds
WKLOAD_RESPONSETIME_MINIMUM	Minimum response time in seconds of completed work units.	seconds
WKLOAD_RESPONSETIME_PERCENTILE	The 90th percentile response time in seconds of units of work that completed.	seconds
WKLOAD_RESPONSETIME_STDEV	Standard deviation of response times in seconds in this interval.	seconds

10.17 XML destination : work_trans_metrics

Table 1: XML destination : work_trans_metrics

XML destination	Metric group	Database staging table
work_trans_metrics	Workload transaction metrics.	work_trans_metrics_staging

10.17.1 Required fields

Table 2: Required fields

Element name	Description
workload_transaction_id	ID of the transaction. This could be same as the transaction name.
workload_transaction_name	Name of the transaction.
workload_transaction_type	Type of transaction such as database transaction, web transaction etc.
metric_time	Time at which metric was reported
gmt_offset	GMT offset in minutes

10.17.2 Optional fields

Table 3: Optional fields

Element name	Description
business_function_name	Name of the IT service. A IT Service may have zero, one or more Sub Systems (Applications) or Business Services. Applications run a host and application metrics can be imported to DM using standard application_metrics_staging table.
business_service_name	Name of the Business Process. A Business process may have zero, one or more business services or IT Services.
duration	Duration of the metric in seconds.
server_id	Host name on which this transaction runs.
subsystem_id	ID of the application that executes this transaction. This value maps to the application_id in the application_metrics_staging table.
subsystem_name	Name of the application that executes this transaction. This value maps to the application_name in the application_ metrics_staging table.
subsystem_type	Type of application. This value maps to the application_type in the application_metrics_staging table.
workload_name	Name of the Business Service. A Business Service may have zero , one or more IT services or Business Processes
workload_type	Type of workload such as IT service or Business service etc.

10.17.3 Standard metrics

Table 4: Standard metrics

Metric name	Metric description	Metric unit
TRANS_CPU_TIME_PER_TRANSACTION	CPU seconds taken per transaction	seconds
TRANS_DB_RESPONSE_TIME	Database response time in seconds	seconds
TRANS_DISK_IO_PER_TRANSACTION	Number of disk IOs per transaction	number of IOs
TRANS_DISK_IO_PER_TRANSACTION	Number of disk IOs per transaction	number of IOs
TRANS_DISK_READBYTES_PER_ TRANSACTION	Number of disk bytes read per transaction	bytes
TRANS_DISK_RESPONSE_TIME	Disk response time in seconds	seconds
TRANS_DISK_TOTALBYTES_PER_ TRANSACTION	Number of disk bytes read and written per transaction	bytes
TRANS_DISK_WRITEBYTES_PER_ TRANSACTION	Number of disk bytes written per transaction	bytes
TRANS_MEM_KB_PER_TRANSACTION	Memory in kilo bytes user per transaction	kilobytes
TRANS_NETWORK_IO_PER_TRANSACTION	Number of network IOs per transaction	number of IOs
TRANS_NETWORK_READBYTES_PER_ TRANSACTION	Number of network bytes read per transaction	bytes
TRANS_NETWORK_TOTALBYTES_PER_ TRANSACTION	Number of network bytes read and written per transaction	bytes
TRANS_NETWORK_WRITEBYTES_PER_ TRANSACTION	Number of network bytes written per transaction	bytes
TRANS_RESPONSE_TIME	Response time of transaction in seconds	seconds

10.18 Metrics used by Capacity Command Center

The following metrics are used by the CA Capacity Command Center product:

10.18.1 Performance Metrics

Server_metrics (for physical servers and hosts)

Metric Name	Metric Description	Metric Unit
GBL_CPU_UTIL	CPU utilization.	percent
GBL_MEM_MB_USAGE	Total memory in megabytes used by the host.	megabytes
GBL_MEM_UTIL	Memory utilization. This value is not used directly by Capacity Command Center. Data Manager uses it to compute GBL_MEM_MB_USAGE.	percent

Server_metrics (for VMs)

Metric Name	Metric Description	Metric Unit
GBL_LS_MEM_MB_USAGE	Total memory in megabytes used by the VM.	megabytes
GBL_LS_PHYS_CPU_UTIL	Logical system physical CPU utilization.	percent
GBL_LS_PHYS_MEM_UTIL	Memory utilization. This value is not used directly by Capacity Command Center. Data Manager uses it to compute GBL_LS_MEM_MB_USAGE.	percent

10.18.2 Configuration data

Server (for physical servers and hosts)

Element name	Description
host_name	Name of the host.
parent_cluster_name	Name of the parent cluster if the host is part of a cluster.

cpu_utilization_type	Type of CPU utilization reported by the monitor. Monitoring agents could report traditional utilization, capacity utilization, purr utilization, core utilization etc. For example, agents running on regular X86 servers report traditional utilization whereas agents running on AIX 5 and 6 systems report PURR utilization.
	Allowed values are
	`0 - Unknown`
	`1 - Regular(Dispatcher) Utilization`
	`2 - Capacity Utilization`
	`3 - Purr Utilization`
	`4 - Core  Utilization`
	`5 - MHz utilization`

num_cores	Number of CPU cores.
num_cpus	Number of CPU threads.
os_type	Operating system.
total_memory	Total physical memory in kilobytes.
virtualization_solution	Name of the virtualization solution such as "VMware ESX 3.5" if the host is a virtualization host.
cores_per_processor_chip	Number of cores per processor chip.
threads_per_processor_core	Number of threads per processor core.

Server_virtual (for VMs)

Element name	Description
server_id	DNS name of the virtual machine. Could be Name or IP address.
physical_host	Name of the physical host on which VM is running.

cpu_utilization_type	Type of CPU Utilization reported by the monitor. Monitoring agents could report traditional utilization, capacity utilization, purr utilization, core utilization etc. For example, agents running on regular X86 servers report traditional utilization whereas agents running on AIX 5 and 6 systems report PURR utilization.
	Allowed values are
	`0 - Unknown`
	`1 - Regular(Dispatcher) Utilization`
	`2 - Capacity Utilization`
	`3 - Purr Utilization`
	`4 - Core  Utilization`
	`5 - MHz utilization`

online_virt_cpu	Number of virtual CPUs in the VM.
online_memory	Entitled memory in kilobytes allocated to VM.
os_type	Operating system running on the VM.

11.0 Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at http://ca.com/docs.

Index

O

P

R

S

T

V

W

X