CA Performance Center

CA Report Information Base API Guide

2.4.1

CA Technologies Product References

This document references the following CA Technologies products:

- CA Performance Center
- CA Infrastructure Management Data Aggregator
- CA Network Flow Analysis
- CA Application Delivery Analysis
- CA NetQoS Unified Communications Monitor

Contact CA Technologies

Contact CA Support

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

- 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

Providing Feedback About Product Documentation

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

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.

Contents

Chapter 1: Introduction

This section contains the following topics:

Custom Reporting with the RIB

CA Performance Center provides multiple built-in reports for monitoring your systems. You can enhance your ability to monitor your systems by creating custom reports. The API lets you build custom reports using the RIB interface.

The *RIB* (Report Information Base) interface is a system of web services and XML documents that lets you access data that is collected from various sources. The RIB system includes the following components:

- RIB Sources—Web services that expose data from different sources.

- RIB Documents—XML files that describe the schema that is exposed by different data sources.

- RIB Engine—An application-facing web service that accepts queries, distributes them among the RIB sources, and combines results for return.

RIB Components

The *RIB* (Report Information Base) interface is a system of web services and XML documents that lets you access data that is collected from various sources. Designed for vast environments, the RIB allows for flexibility and a broad scope for reporting of data from monitoring and management platforms.

The RIB system provides a SQL-like interface for querying data. It includes the following components:

RIB Sources

RIB sources are web services that encapsulate the details of exposing data from a particular source. Virtually any source of data can be packaged as a RIB source.

RIB Documents

RIB documents are XML documents that describe the capabilities of the data sources.

RIB Engine

The RIB engine is an application-facing web service that accepts queries, distributes them among RIB sources, and combines results for return.

RIB uses web services as a communications framework, which distributes processing workload among separate components. The RIB standard provides a single web service signature for use with various sources, enabling client code to work with a variety of data sources.

The following illustration shows how RIB components interact:
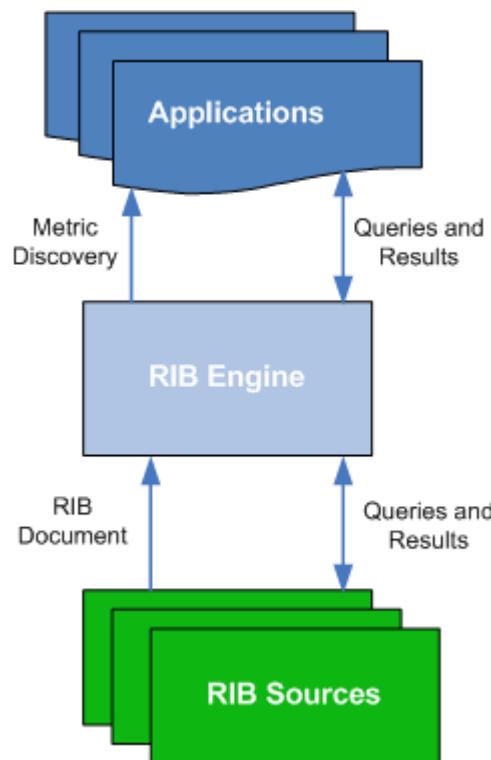
The RIB Engine

The *RIB engine* is an application-facing web service that accepts queries, distributes them among RIB sources, and combines results for return.

Although the database RIB queries resemble SQL queries, the RIB engine is not a SQL execution engine. Where an execution engine parses a query and makes various calls to source functions according to the query execution strategy, the RIB engine passes entire queries to the appropriate data source for execution. All aggregation, filtering, sorting, and other database functions must be implemented at the data source level. Otherwise, they are not supported.

Custom queries are therefore subject to data source limitations. For such queries, each data source must explicitly opt into supporting what it can reasonably support by including a statement in its RIB document. Any custom queries that would overtax the data source can be excluded in this manner.

For more information, see RIB Documents (see page 10).

The RIB engine does perform certain operations on datasets when a RIB query spans multiple data sources. For example, the engine combines results across multiple sources of the same type.

How the RIB Engine Gets Data

When a new data source is registered with CA Performance Center, the following occurs:

1. A web service on the data source reports whether it supports the RIB interface.

2. The data source reveals the URL of its RIB source web service.

3. The RIB engine queries CA Performance Center for a list of data sources supporting the RIB.

4. The RIB engine pulls a list of RIBs supported by each data source, and will federate any incoming queries to the appropriate sources.

After registration, CA Performance Center synchronization with the data source normally occurs every 5 minutes. This recurring, automatic synchronization consists of database updates from all data sources, propagation of configuration changes from the CA Performance Center Admin interface to the data sources, and reconciliation of new data within the CA Performance Center database.

Some data sources may frequently update their RIB document based on current data. During synchronization, the RIB engine queries the RIB document for updated definitions and forwards them to CA Performance Center. Shared definitions are propagated to all registered data sources.

RIB Documents

The RIB system is designed to avoid excessive processing load on CA Performance Center and its underlying data sources. One example of this design is the way that individual data sources "opt in" to capabilities in a standardized document. A *RIB document* is an XML document that describes the capabilities of a data source. The RIB document also defines all of the valid ways of querying a particular RIB source. Operations that would create excessive load on the data source are prevented by being excluded from the RIB document.

A RIB document resembles an SNMP MIB, describing *types* and *tables*. XML statements detail the capabilities of the data source and include metadata that further describes the data that the source provides.

Like MIB objects, RIB elements are associated with hierarchical namespaces. As a result, these elements can be used to organize RIBs into a hierarchical structure that can be walked by clients.

The RIB document also includes information specifying how to display the results of a given query. For example, a field may be of the type 'milliseconds'; *milliseconds* can be abbreviated as *ms*; 1000 milliseconds is a second; etc. Such rules are not within the purview of CA Performance Center but are enforced by the data source by means of the RIB document. As a result, each data source has an increased ability to define appropriate types of data and units.

RIB Document Contents

RIB documents declare information about data source capabilities by exposing a "logical schema" that reveals the types of data and the RIB tables that are supported. RIB documents also describe the appropriate means of presenting the data in reports.

The RIB tables in a RIB document describe the data available from a given source. These tables can abstract federated databases, partitioning schemes, or other data storage patterns that can present challenges for an application seeking to use the data. A RIB Table declaration includes:

- a list of fields
- a list of valid ways that the fields in the table can be aggregated ("groupings")
- a list of possible related tables ("table joins")

RIB fields resemble database fields, but they can take parameters, and they can be organized hierarchically. RIB tables list the valid arrangements of fields to be used in grouping data. Where SQL allows grouping on arbitrary sets of fields, the RIB acknowledges that underlying data sources may only provide data aggregated in a limited number of ways. Therefore, data sources must explicitly opt into the various supported combinations.

The SQL concept of joining database tables is deployed within the RIB as an implicit join. *Implicit joins* are invoked by referring to fields in a foreign table as if they were part of the table being queried. Implicit joins occur within a single RIB source.

Finally, the RIB document must also contain the following items:

Version number

Can be checked on the client side to ensure compatibility with a particular application.

Included RIBs

Lets you deploy a common set of types among multiple RIB sources, which are then referenced by each other's RIB documents.

Unit definitions

Define units that can be referenced by fields within RIB tables. Unit definitions are used by clients to display query results.

Chapter 2: RIB Syntax

The syntax for RIB queries resembles that of SQL queries, with some differences. RIB queries conform to the following structure:

- SELECT [field list]

- FROM [table]

- WHERE [logic expression]

- GROUPBY [field list] [optional]

- ORDERBY [field list]

- LIMIT [number of records]

- OFFSET [number of records]

The where clause is supported in a manner similar to SQL. Likewise, the RIB specification supports the typical database operators (: >, <, >=, <=, =, LIKE, and IN); however, individual data sources may not support all of these.

RIB queries identify tables and fields by a fully qualified name consisting of the following: the name of the RIB that the table or field belongs to; the table name; and the field name, if any. Field names in queries are commonly abbreviated by using relative names, indicated by a leading '.' before the field name. Because queries specify a table according to its fully qualified name, fields referenced in the same query can be assumed to be within that table. For example,

```
SELECT .ItemName from CA.DataSourceName.PollItem
```

is equivalent to

```
SELECT CA.DataSourceName.PollItem.ItemName from CA.DataSourceName.PollItem.
```

Another difference between RIB syntax and SQL syntax lies in referring to related tables. In SQL, the syntax requires explicitly joining the two tables. In the RIB system, the mechanics of the join are hidden from the end user. As a result, fields from related tables can be referred to directly. For example, if the InterfaceStats table is related to the PollItem table, we can refer to any field from PollItem in a query against InterfaceStats. The following syntax retrieves the item name and IP address associated with each record in the InterfaceStats table in addition to the count of bytes in:

```
SELECT .PollItem.ItemName, .PollItem.IPAddress, .BytesIn from
CA.DataSourceName.InterfaceStats
```

This section contains the following topics:

Syntax for Field Groups

RIB syntax lets you create field groups, in which several subfields are grouped under a single name. This strategy is useful for exposing variations on fields, such as minimum, maximum, or average data values. The field group, in this case, might be **.UtilizationIn**, and the whole field names would be **.UtilizationIn.Avg**, **.UtilizationIn.Max**, and so on.

```
<FieldGroup id=".UtilizationIn" aggregate="true">
        <Properties>
         <Property name="TypeRef">Global.Percent</Property>
         </Properties>
     <FieldRef id="CA.DataSourceName.Templates.Max" />
     <FieldRef id="CA.DataSourceName.Templates.Avg" />
```

Syntax for Table Joins

Relationships between tables are declared in the tablejoin elements in the RIB document.

```
<TableJoin id="CA.DataSourceName.Device" alias=".Device">
```

This statement declares that all fields from the Device table are available within the current table under the '**.Device**' alias. If this statement appeared in the InterfaceStats table, you could then do something like the following:

```
Select .Device.IPAddress, .Device.Name, .InOctets from
   CA.DataSourceName.InterfaceStats
```

In some cases, a table may be joined two or more times to another table. In such cases, the table alias is different for each join:

```
<TableJoin id="CA.DataSourceName.Device" alias=".Client">
<TableJoin id="CA.DataSourceName.Device" alias=".Server">
```

Queries would then look something like this:

```
Select .Client.IPAddress, .Client.Name, .Server.IPAddress, .Server.Name,
   .InOctets from CA.ReporterAnalyzer.Conversations
```

Syntax for Groupings

Groupings express constraints on which fields can be used together in queries that require aggregation. For example, a RIB source may be able to aggregate volume by interface or by interface and protocol, but not by protocol alone. These acceptable combinations are listed in the <Groupings> section of a table declaration in the RIB document. The following is an example:

```
<Groupings>
    <GroupBy unique="true">
      <FieldRef id=".EndTime"/>
    </GroupBy>
    <GroupBy unique="true">
      <TableRef id=".PollItem"/>
    </GroupBy>
    <GroupBy unique="true">
      <TableRef id=".PollItem"/>
      <FieldRef id=".EndTime"/>
    </GroupBy>
```

Chapter 3: Using the RIB API

This section contains the following topics:

Access the RIB API

The CA Performance Center RIB API provides RIB document access in XML format.

Follow these steps:

1. Log into the target computer as root, or use a remote program, such as putty, to communicate with a remote computer.

    Note: If you do not have root access, use an account with sudo privileges. For more information, see the *CA Performance Center Installation Guide*.

2. Open a web browser.

3. In the address field, enter the following address:

    ```
    http://<server IP address>:8481/dm/rib/doclist
    ```

    where:

    <server IP address>

    Is the IP address of the computer where you installed the CA Performance Center software.

    8481

    Is the port number where the Device Manager daemon listens for requests.

    The browser displays the RIB API index page.

Source Restrictions

By design, the RIB document that the data source presents should completely describe the valid ways of querying a source. However, in certain cases, source restrictions cannot be thoroughly described.

For example, a data source requires that time bounds be used in the 'where' clause of queries against statistics tables. The syntax must include a clause for '.EndTime > [value]' and '.EndTime < [value]'.

In some cases, such restrictions must be documented externally.

Chapter 4: Reporting Interface Base (RIB) Web Services

This section contains the following topics:

RIB Interface Web Service

A RESTful web service lets you run queries to the RIB engine. Use this web service to test data source RIB documents.

URI and Operations

Base URI = http://<Server IP address>:8481/dm/rib/

The RIB web service offers the following three methods:

doclist

Gets the names and versions of all RIB documents supported by the RIB source being queried. Typically, these include a primary document that outlines the tables and fields supported by the data source and any RIB documents that the primary document refers to by means of the **ReportInformationBaseRef** statement.

Example: http://133.233.33.133:8481/dm/rib/doclist

doc

Returns the actual RIB document when given a RIB document identifier as a parameter. Lets consumers discover all of the information necessary to issue queries to a selected RIB source.

Example: http://133.233.33.133:8481/rib/doc/CA.*DataSourceName*

query

Runs a RIB query and returns results (data).

Example: http://133.233.33.133:8481/dm/rib/query/select .ItemName, .ItemDescription from CA.*DataSourceName*.PollItem

Examples

doclist

```
<ribdocuments xmlns="http://im.ca.com/portal/rib/link"
xmlns:ns2="http://im.ca.com/portal/rib/doclist">
    <ribdocument id="CA.IM.DA" version="1.0.1">
        <link rel="document" href="http://localhost:8481/dm/rib/doc/CA.IM.DA"/>
    </ribdocument>
    <ribdocument id="CA.IM.DA.MF.NormalizedAvailabilityInfo" version="1.0.1">
        <link rel="document"
href="http://localhost:8481/dm/rib/doc/CA.IM.DA.MF.NormalizedAvailabilityInfo"/>
    </ribdocument>
    <ribdocument id="CA.IM.DA.MF.NormalizedCPUInfo" version="1.0.1">
        <link rel="document"
href="http://localhost:8481/dm/rib/doc/CA.IM.DA.MF.NormalizedCPUInfo"/>
    </ribdocument>
    ...
    ...
</ribdocuments>
```

doc

```
<?xml version="1.0" encoding="utf-8" standalone="no" ?>
<ReportInformationBase id="CA.DataSourceName" version="1.0.0">
  <ReportInformationBaseRef id="NetQoS" />
  [RIB document returned here]
```

query

Select ItemName, ItemDescription from PollItem

Example of Query Syntax: Top N Memory

http://133.233.33.133:8481/dm/rib/query/select .PollItem.PollItemID, .CPU_UTIL from CA.*DataSourceName*.RTR_CPUSTATS where .EndTime(300) > 1299697838 and .EndTime(300) < 1299699640 groupby .PollItem.PollItemID orderby .CPU_UTIL desc

Appendix A: Sample RIB Queries

This section contains the following topics:

Query the RIB to Return a View with Business-Hour Filtering

You can query the RIB to return data for a specific metric by entering queries into a web browser. This example presents a CA Performance Center RIB query that returns Top Discards data from a Data Aggregator data source.

Precede all CA Performance Center RIB queries with the following URL:

```
http://<server IP address>:port/dm/rib/query/
```

server IP address:port

> Is the IP address of the computer where you installed the CA Performance Center software and the port number where the Device Manager daemon listens for requests.

> **Default port**: 8481

Note: For information about submitting a query directly to a Data Aggregator data source, see the *CA Infrastructure Management Data Aggregator RIB API Guide*.

You can append URL parameters to specify property values:

```
http://<server IP address>:port/dm/rib/query/ribquery
/?property1=value1&property2=value2
```

The following RIB query returns Top Discards data from a Data Aggregator data source:

```
http://<server IP address>:port/dm/rib/query/SELECT .PollItem.ID,
.PollItem.DevDisplayName, .Item.DisplayName, .Discards.Sum, .DiscardsIn.Sum,
.DiscardsOut.Sum FROM CA.IM.DA.MF.NormalizedPortInfo.IFSTATS WHERE
.Group.GroupID = 1039 AND .EndTime(300) > 1366208760 AND .EndTime(300) <=
1366212360 GROUPBY .PollItem.ID, .Item.DisplayName, .PollItem.DevDisplayName
ORDERBY .Discards.Sum DESC LIMIT 10
```

Tip: If necessary, you can escape the RIB query and parameters. Many web browsers escape the RIB query as follows:

```
http://<server IP
address>:port/dm/rib/query/SELECT%20.PollItem.ID,%20.PollItem.DevDisplayName,
%20.Item.DisplayName,%20.Discards.Sum,%20.DiscardsIn.Sum,%20.DiscardsOut.Sum%
20FROM%20CA.IM.DA.MF.NormalizedPortInfo.IFSTATS%20WHERE%20.Group.GroupID%20=%
201039%20AND%20.EndTime(300)%20%3E%201366208760%20AND%20.EndTime(300)%20%3C=%
201366212360%20GROUPBY%20.PollItem.ID,%20.Item.DisplayName,%20.PollItem.DevDi
splayName%20ORDERBY%20.Discards.Sum%20DESC%20LIMIT%2010
```

Add the following URL parameters to return Top Discards data for a set of business hours in a specific time zone. Sets of business hours are configured by a CA Performance Center administrator.

Note: Not all queries support data filtering by time zone and business hours.

RIB.TimeZone

> Is the string identifier of the time zone used to filter data results.

RIB.BusinessHours

> Is the CA Performance Center ID of the business hour definition used to filter data results. Include this parameter in the propertiesToTranslate value to ensure that the ID is translated. IDs that are not translated are submitted unchanged to each applicable data source.

propertiesToTranslate

> Is a list of parameter names whose values contain a CA Performance Center ID to translate to a local data source ID.

Example 1

To return data filtered by time zone, add the time zone parameter (shown in bold text) to the URL. In the following example, the data is filtered to include only data for items in sites configured for the America/New_York time zone.

```
http://pchost:8481/dm/rib/query/SELECT .PollItem.ID, .PollItem.DevDisplayName,
.Item.DisplayName, .Discards.Sum, .DiscardsIn.Sum, .DiscardsOut.Sum FROM
CA.IM.DA.MF.NormalizedPortInfo.IFSTATS WHERE .Group.GroupID = 1039 AND
.EndTime(300) > 1366208760 AND .EndTime(300) <= 1366212360 GROUPBY .PollItem.ID,
.Item.DisplayName, .PollItem.DevDisplayName ORDERBY .Discards.Sum DESC LIMIT
10?RIB.TimeZone=America/New_York
```

Example 2

To return data filtered by time zone and business hours, add the time zone and business hours parameters (shown in bold text) to the URL. In the following example, the data is filtered to include only data for items in sites configured for the America/New_York time zone and business hours matching the CA Performance Center definition for ID 6434.

```
http://pchost:8481/dm/rib/query/SELECT .PollItem.ID, .PollItem.DevDisplayName,
.Item.DisplayName, .Discards.Sum, .DiscardsIn.Sum, .DiscardsOut.Sum FROM
CA.IM.DA.MF.NormalizedPortInfo.IFSTATS WHERE .Group.GroupID = 1039 AND
.EndTime(300) > 1366208760 AND .EndTime(300) <= 1366212360 GROUPBY .PollItem.ID,
.Item.DisplayName, .PollItem.DevDisplayName ORDERBY .Discards.Sum DESC LIMIT
10?RIB.TimeZone=America/New_York&RIB.BusinessHours=6434&propertiesToTranslate
=RIB.BusinessHours
```

Glossary

implicit join

Implicit joins are invoked by referring to fields in a foreign table as if they were part of the table being queried. Implicit joins occur within a single RIB source.

RIB

The *RIB* (Report Information Base) interface is a system of web services and XML documents that lets you access data that is collected from various sources.

RIB document

A *RIB document* is an XML document that describes the capabilities of a data source. The RIB document also defines all of the valid ways of querying a particular RIB source.

RIB engine

The *RIB engine* is an application-facing web service that accepts queries, distributes them among RIB sources, and combines results for return.

RIB sources

RIB sources are RESTful web services that encapsulate the details of exposing data from a particular source. Virtually any source of data can be packaged as a RIB source.