Report optimization for MDX queries

Reports retrieve numeric data from the application database through MDX queries. The way a report is defined determines the MDX queries generated and can have a substantial impact on the refresh performance of a report. Each MDX query has a timeout to prevent run away queries on the server.

Timeouts – each MDX query executed for a report has a timeout to prevent run away queries on the server. This time is set in the Modeler on the Server Setup page. In the Web Service Settings section, the Query Timeout is the timeout setting in seconds. An Administrator can use this setting to control the maximum time any query is allowed to execute on the server. The default setting is 30 seconds. If your reports get an xmla timeout error, this is the parameter to change. This is also the way an Administrator limits just how much of the shared server query resources are available at any point in time to individual users.

How a report builds MDX queries

For a simple report with all rows and columns at the top level and no nesting as shown below, one MDX query is generated for all combinations of all columns by all rows.



When the report gets more complex, as shown below, this can result in a query to retrieve much more data than required to satisfy the defined report. This report displays all months of Actual and only the current period for Budget. However as currently defined the report generates one query to retrieve all requested time periods by scenarios actual and budget by all requested accounts.



When a report contains rows or columns defined at more than just the top level, each of the nodes with a nesting level more than one generates additional more specific queries. So the same report as above can be modified to produce the same results, but very different queries for data. Notice that the last column in the definition now contains a mbr:Time with the mbr:Scenario nested below it. Now the last column is not included in the top level query for the report, but rather generates a smaller query for just what is needed for the last column. So now the report generates 2 MDX queries. The first query is all requested time periods by scenario actual by all requested accounts . The second query is one time period by scenario budget by all requested accounts. So with this simple change to the report all time periods for the budget scenario are no longer requested in the first query, making that query lighter, and the second query only asked for one time period for the budget scenario. This same optimization applies to rows as well as columns.

Using this technique can have dramatic results on the performance of complex reports.

