Intelligent Sparse Reporting (ISR)

This document explains how you can improve your ShowCase® Essbase® applications with Intelligent Sparse Reporting (ISR). Here's what you'll find in this document:

- Overview
- About Intelligent Sparse Reporting
- Implementing ISR For Your Application and Database
- Activating and Deactivating ISR
- ISR Components
- Troubleshooting
- Frequently Asked Questions

Overview

Intelligent Sparse Reporting (ISR) is a licensed enhancement to ShowCase Essbase that enables faster query/retrieval processing for Essbase databases containing related dimensions comprised of a large number of members (defined as sparse dimensions to the Essbase engine). As an example, a database may contain several dimensions representing product SKU's such as Product, Color, and Size.

Although there may be a very large number of Color and Size members, the number of members relating to a Product or set of Products may be very small. If the database meets the criteria, then ISR can greatly enhance the retrieval time for any front-end product retrieval, such as Analyzer, Query, Report Writer, or a spreadsheet that queries the Essbase database.

In many cases, a retrieval/query that causes Essbase to search millions of "blocks" can be optimized to search only a few thousand or even a few hundred blocks, reducing the amount of time and resources required to satisfy the request. In essence, ISR optimization uses a set of user-provided rules or constraints identifying valid member combinations to optimize a report focusing only on data that exists, eliminating unnecessary searching and processing.

Users could obtain this same benefit by optimizing individual reports manually. However, the process to manually generate this report can be time-consuming and problematic due to the dynamic nature of multidimensional databases.

Can Your Organization Benefit from ISR?

Not every Essbase application will benefit from ISR. It is important to identify applications that have a profile that will allow ISR to improve performance. Here are guidelines to help you determine whether your application/database is a good candidate for the benefits ISR provides.

- Do you generate mission-critical reports that simply take too long or consume too many resources while processing?
- Would you like to reduce retrieval time for spreadsheets, Analyzer, or other front-end tools that query your Essbase cubes?
- Do your reports use any of the following report writer commands: ALLSIBLINGS, ALLINSAMEDIM, (I)CHILDREN, (I)DESCENDANTS, DIMBOTTOM, ONSAMELEVELAS? Keep in mind that equivalent commands are generated when using spreadsheets, Analyzer, or other front-end tools. For instance, drilling down on a category member within a Products dimension may result in the equivalent of doing a DIMBOTTOM.
- Will the Report Writer commands reference members in a sparse dimension (i.e., the (I)CHILDREN of Products)? See the outline in "How ISR Works" on p. 3.
- Does the report return large amounts of data with #missing or are you suppressing #missing return values?
- Can you identify relationships between multiple dimensions in your database? These may be related to a single entity as in the example of a product SKU broken into component parts. These may also be relationships that define "business logic," such as certain products only purchased by a certain group of customers.

General Considerations

Asymmetric Reports. ISR will not provide performance improvements for Asymmetric reports. See the Hyperion Essbase documentation for more information about Asymmetric reports.

Report Script Commands. The following Report Writer commands are not supported: RESTRICT, TOP, BOTTOM, ORDER BY, DIMEND, WITHATTR, LINK*****. If a report is run that contains these functions, the report will run as normal, but there will be no performance improvements from ISR. In addition, when using ONSAMELEVELAS against a non-level 0 member, the report will run as normal, but there will be no performance improvements from ISR.

******Note*: Actions in other programs might generate these commands. For example, if "Does not exist at level 1" is chosen in the subset member selection in Analyzer, or "Member Filter" is used in Excel Query Designer, the LINK function will be used to retrieve the date from Essbase.

Explicitly Stated Members. Member names that are explicitly stated in a report will be displayed in output.

Restoring the Database. If a database is restored, the timestamp will be reset. This will invalidate the rules. You will have to delete the rule and then recreate it with the CRTOLAPRUL command.

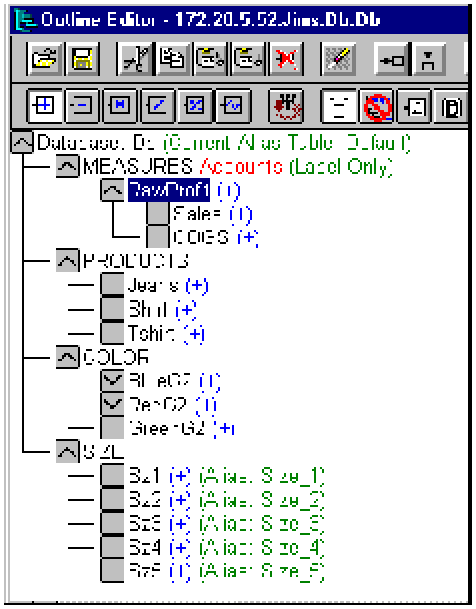
Double-Byte Character Sets. Currently, ISR does not support DBCS.

About Intelligent Sparse Reporting

How ISR Works

Intelligent Sparse Reporting gives ShowCase Essbase reporting additional run-time intelligence based on a *rule.* As you read about ISR, you will notice references to “valid member combinations.” That’s because you define a rule by specifying valid member combinations for a set of selected sparse dimensions. At report execution time these valid member combinations are used to create a script that searches for and retrieves only the data which has valid member combinations.

The following outline may help explain how the Intelligent Sparse Reporting feature works.



Outline Editor - 172.20.5.52.Jims.Db.Db

```
Database: Db (Current Alias Table: Default)
    MEASURES Accounts (Label Only)
        RawProfit (~)
            Sales (~)
            COGS (+)
    PRODUCTS
        Jeans (+)
        Shirt (+)
        Tshirt (+)
    COLOR
        Blue02 (~)
        Red02 (~)
        Green02 (+)
    SIZE
        Sz1 (+) (Alias: Size_1)
        Sz2 (+) (Alias: Size_2)
        Sz3 (+) (Alias: Size_3)
        Sz4 (+) (Alias: Size_4)
        Sz5 (~) (Alias: Size_5)
```

From the outline:

- ■Assume that dimensions PRODUCTS, COLOR, and SIZE are all sparse dimensions.
- ■Assume that jeans are only sold in the color blue.

Without ISR, a typical report script against this outline may search all possible color and size combinations for product jeans, even though jeans are only sold in blue. All other color and size combinations could be eliminated if the user supplied the valid color and size combinations for all existing products.

ISR enables you to specify the valid COLOR and SIZE member combinations for a PRODUCT to provide the rules for this outline.

```
// Unoptimized Report for RawProfit Jeans
 {DECIMAL 2}
//
<PAGE (MEASURES, PRODUCTS, SIZE)
   RawProfit   Jeans
{OUTALTNAMES}
<COLUMN ("SIZE")
<ICHILDREN SIZE
// Show all member Colors in Dimension
 <ROW (COLOR)
 <CHILDREN COLOR
```

A typical report such as this would cause Essbase to search a combination of sizes, products, and colors because it cannot predict which combinations have no values.

Implementing ISR for Your Application and Database

Following are the general steps required to activate and deactivate ISR for a single application/database. For more detail on the individual command parameters, see “Activating and Deactivating ISR” on p. 4.

Activating ISR

- ▶ **Step 1**: Determine if the application/database is a good candidate for ISR. See “Overview” on p. 1.
- ▶ **Step 2**: Identify the sparse Dimensions that are relevant to your reporting.
- ▶ **Step 3**: Write a SELECT statement that will effectively identify only valid member combinations for the dimensions identified in step 2. See user SELECT statement information under “CRTOLAPRUL Command” on p. 4.
- ▶ **Step 4**: Run the CRTOLAPRUL command specifying the dimensions and SELECT statement from steps 2 and 3, respectively. See “CRTOLAPRUL Command” on p. 4.

Deactivating ISR

Use the DLTOLAPRUL command to deactivate or end optimization for a single application/database. See “DLTOLAPRUL Command” on p. 7.

Activating and Deactivating ISR

The two iSeries commands associated with the ISR enhancement activate and deactivate optimization for a specific application/database. The CRTOLAPRUL command creates the necessary ISR objects required for optimization, and the DLTOLAPRUL command deletes or deactivates optimization for a specific application/database.

CRTOLAPRUL Command

Run CRTOLAPRUL to create the necessary objects and activate report optimization. The command allows you to specify valid member combinations from selected sparse dimensions. These valid member combinations create the Rules Optimizer Table (ROT). The command creates one ROT per application/database specified. To change a rule for an application/database, delete the rule (DLTOLAPRUL), and then create a new one with CRTOLAPRUL.

The command requires a valid SELECT statement as one of its parameters. The purpose of the user-supplied SELECT statement is to feed Level 0 member names to the program that creates the Rules Optimizer Table. The complete set of valid member combinations, one member for each dimension, constitutes a rule. As each row is read from the SELECT statement, the ROT is populated with a corresponding numeric row that can be used at run-time to optimize retrievals/queries.

Each column in the SELECT statement should yield a Level 0 member name that exists in the outline. This does not mean the SELECT statement must contain a row for every combination of sparse members. You are not required to include each sparse dimension in the definition, but must include a row for each combination of the dimensions identified in the CRTOLAPRUL command. It is important to ensure that the SQL statement performs any necessary modification to the relational data representing the level 0 Essbase member so that the result will match the loaded member name.

Notes:

- The member names are often altered as they are loaded into Essbase to satisfy Essbase's unique member name requirement. Each column is processed according to the order of the dimensions specified in DIMRLIST on the CRTOLAPRUL command.
- The order of the dimensions in the DIMRLIST and the selected columns must match. For example, if Products is specified as the first dimension on the DIMRLIST parameter, then column one of the returned result set from the SELECT statement must contain member names for the Products dimension. Any member names that do not exist in the Products dimension will be rejected and logged.
- You cannot use OLAP rules for outlines with non-zero level members that are referred to by a shared member. Attempting the CRTOLAPRUL command on such an outline results in message ESA8027, which reads:

  Outline not eligible for OLAP rules. Member &1 is a non-zero level member which is referred to by a shared member.

Performance Tips

When the ROT is created, the relationship between the dimensions determines how effectively ISR will improve performance. The SELECT statement must yield DISTINCT values for the valid member combinations. The number of valid combinations loaded as a result of the user SELECT statement should be extremely sparse. If there is an entry for every member combination of the dimensions specified on the CRTOLAPRUL command, then performance will not improve since there is little sparsity.

Valid SQL Data Types

When the SELECT statement is coded, the resulting columns must be either SQL_CHAR, SQL_VARCHAR, SQL_INTEGER, or SQL_SMALLINT; or you can use the SQL function CAST to make these data types compatible with ISR.

Syntax

```
essbaselib/CRTOLAPRUL SVRUSER(userid) SVRPW(password) ROBJLIB(rotlib)
ERRFILE(errorlib/errorfilename) APPNAME(applicationname) DBNAME(databasename)
DIMRLIST(dim01 dim02 dim03)
SQLSTMT('Select dim01, dim02, dim03 from library.tablename')
```

Where:

essbaselib is the name of the ShowCase Essbase installation library.

userid is the user id you normally use to access the Essbase Server.

password is the password you normally use to access the Essbase Server.

rotlib is the library name where you want ISR to create the Rules Optimizer Table.

errorlib is the library name where you want to store the error file.

errorfilename is the name of the error file to log the members rejected when the ROT is created. If you specify an existing error file, the CRTOLAPRUL will fail.

applicationname is the name of the application you want to associate with the OLAP rule you are creating.

databasename is the name of the database you want to associate with the OLAP rule you are creating.

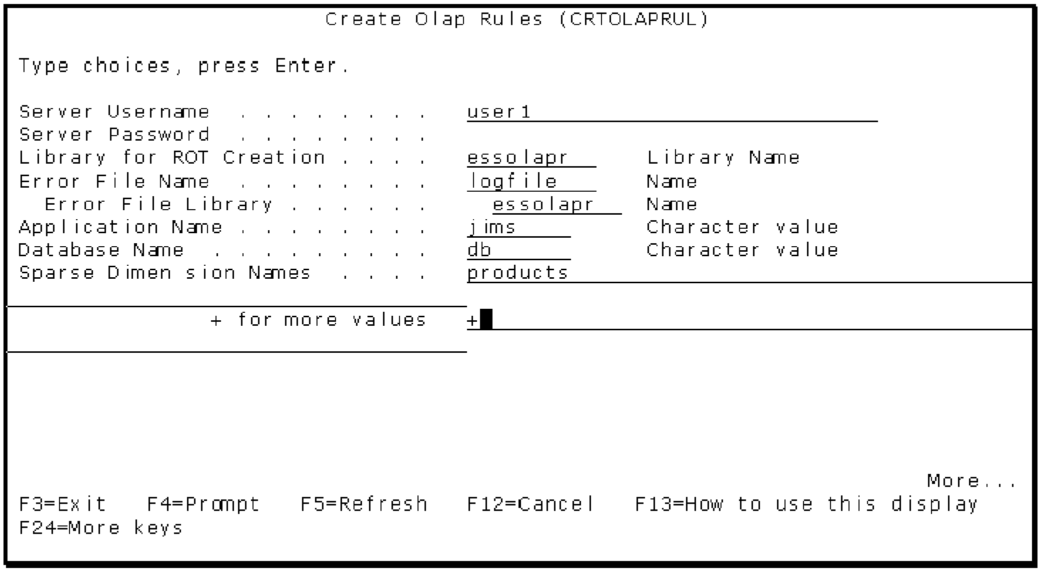
dim01 dim02 dim03 are examples of the sparse dimension names to specify valid combinations. The sequence of the sparse dimension names in DIMRLIST must match the sequence of the sparse dimension names in SQLSTMT. See “Valid SQL Data Types” on p. 5.

library.tablename is the library and table where the members exist. The table must be local. That is, it must be on the same system where ISR is installed.

CRTOLAPRUL Screens

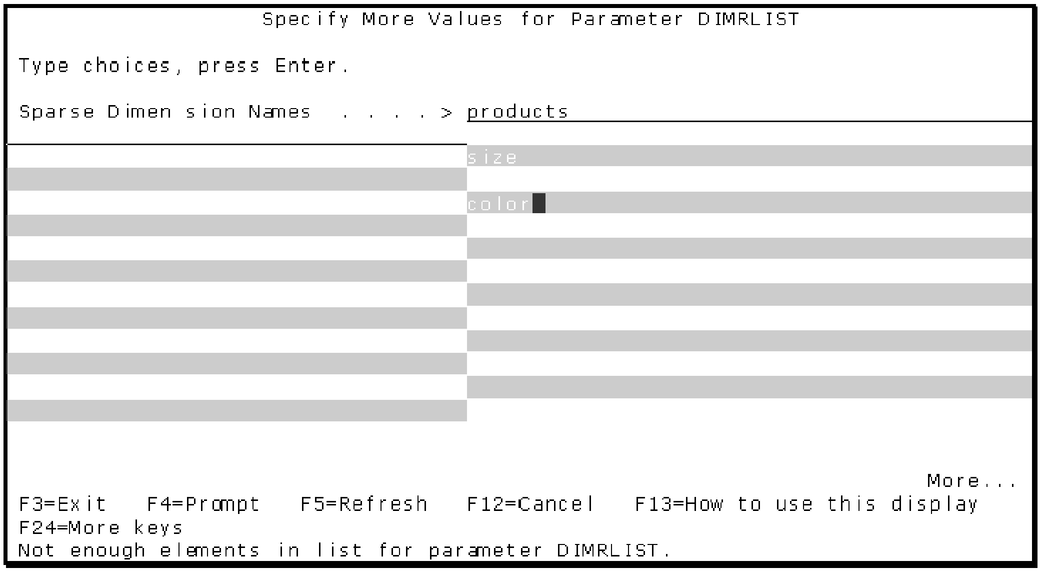
Here are the prompt (F4) screens when running the CRTOLAPRUL command:

Figure 1
CRTOLAPRUL Screen #1



```
                          Create Olap Rules (CRTOLAPRUL)

 Type choices, press Enter.

 Server Username  . . . . . . . .   user1
 Server Password  . . . . . . . .
 Library for ROT Creation . . . .   essolapr     Library Name
 Error File Name  . . . . . . . .   logfile      Name
   Error File Library . . . . . .     essolapr   Name
 Application Name . . . . . . . .   jims         Character value
 Database Name  . . . . . . . . .   db           Character value
 Sparse Dimension Names . . . . .   products
                + for more values   +

                                                                     More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
```

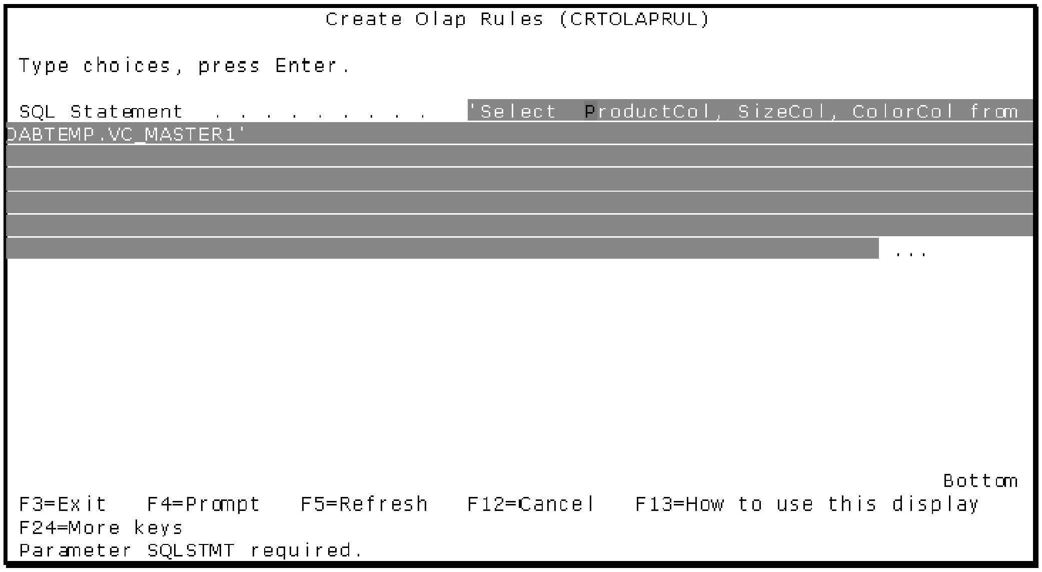
Figure 2
CRTOLAPRUL Screen #2



```
                          Specify More Values for Parameter DIMRLIST

 Type choices, press Enter.

 Sparse Dimension Names  . . . . > products
                                  size

                                  color
```

```
                                                                        More...
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 Not enough elements in list for parameter DIMRLIST.
```

Figure 3
CRTOLAPRUL Screen #3



```
                               Create Olap Rules (CRTOLAPRUL)

 Type choices, press Enter.

 SQL Statement  . . . . . . . . . .  'Select  ProductCol, SizeCol, ColorCol from
DABTEMP.VC_MASTER1'
                                                                            ...
```

```
                                                                         Bottom
 F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
 F24=More keys
 Parameter SQLSTMT required.
```

DLTOLAPRUL Command

This command disassociates the rule from a specific application and database. Since you cannot change a rule, you must delete the rule with DLTOLAPRUL and then use CRTOLAPRUL to resubmit the rule. Information required to delete a rule exists in the catalogs that are updated as the rule is created.

Deleting a rule entails the following:

- SQL Rules Optimizer Table is removed from the library where it exists
- All catalogs are updated to remove references to the rules objects

Note: DLTOLAPRUL does not delete the error file. Running CRTOLAPRUL against an application/database that has an existing error file will fail.

Syntax

essbaselib/DLTOLAPRUL APPNAME(*applicationname*) DBNAME(*databasename*)

Where:

essbaselib is the name of the ShowCase Essbase installation library.

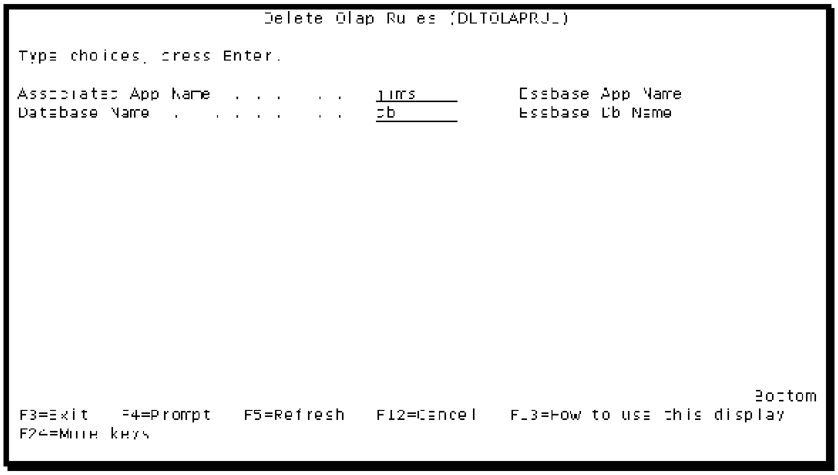
applicationname is the name of the application you want to associate the OLAP rule you are creating.

databasename is the name of the database you want to associate the OLAP rule you are creating.

DLTOLAPRUL Screen

Figure 4 is an example of what you will see if you use prompting (F4) to run the DLTOLAPRUL command:

Figure 4
DLTOLAPRUL Screen #1



```
                         Delete Olap Rules (DLTOLAPRUL)

Type choices, press Enter.

Associated App Name  . . . .   . .   jims          Essbase App Name
Database Name  .   . . . . .   . .   db            Essbase Db Name















                                                                         Bottom
F3=Exit   F4=Prompt   F5=Refresh   F12=Cancel   F13=How to use this display
F24=More keys
```

ISR Components

This section summarizes the iSeries components for the Intelligent Sparse Reporting feature.

Table 1
ISR Components

Object Name	**Object Type**	**Object Description**
CRTOLAPRUL	CMD	Specifies the valid combinations for a specific application/ database. Creates necessary objects prior to report optimization.
DLTOLAPRUL	CMD	Deletes a rule for a specific application/database. This will not affect the Essbase outline. *Note*: The error file does not get deleted with this command. CRTOLAPRUL will fail if the application/ database already has an error file associated.
Rules Optimizer Table (ROT) One instance per application/ database	SQL Table	**DO NOT MODIFY!** The CRTOLAPRUL command creates one instance of this table for each application/database. It is populated with numeric mappings that reflect the sparse member combinations derived from the outline and sparse dimension list specified in the CRTOLAPRUL command.
ROTCATALOG (Rules Optimizer Table Catalog)	SQL Table	**DO NOT MODIFY!** A ShowCase-created table that registers the existence of a ROT after successful creation.
ROTDIMCAT	SQL Table	**DO NOT MODIFY!** A ShowCase-created table that records Dimension information taken from the DIMRLIST parameter on the CRTOLAPRUL command.

Troubleshooting Intelligent Sparse Reporting

When the SELECT statement is processed to build the ROT, a selected row is rejected if any member in the SELECT columns is not a valid Level 0 member in the outline. If any single member in a selected row is rejected, the entire row is logged to the error file specified on the CRTOLAPRUL command. Each member is logged to the error file in the same order presented in the SELECT statement and tab-delimited. This format allows the user to easily view and process the file if they choose.

Typical Reasons for Rejected Rows

Sequencing. The columns in the SELECT statement are sequenced differently than the dimensions specified on the CRTOLAPRUL command. Consequently, the wrong dimensions are searched for members.

Case Sensitivity. The outline is case sensitive, but the SQL feeds variations that have member names that are in mixed case (i.e., ‘Small’ and ‘small’).

Case Sensitivity

The default for Essbase outlines is **NOT** case sensitive. In this case, when the ROT table is built, the member names are normalized internally and case sensitivity is ignored. Sizes ‘Small,’ ‘SMALL,’ or ‘small’ would be considered the same.

Example:

- Assume the Dimensions listed on the CRTOLAPRUL file are Product, Size, and Color
- Assume the SELECT statement is 'Select Products, Size, Color FROM MYLIB.VALIDCOMBOS’

Rejected records in the error file may appear as follows:

Actual logged error

```
---------------------------------------------

BogusProduct•Sz3•Red                      (member for Product was rejected)

Pants•BogusSz•Green                       (member for Size was rejected)

T-Shirt•Sz5•BogusColor                    (member for Color was rejected)

BoguseProduct•BogusSz•BogusColor          (all members in the row were rejected)
```

Shared Members

You cannot use OLAP rules for outlines with non-zero level members that are referred to by a shared member. Attempting the CRTOLAPRUL command on such an outline results in message ESA8027, which reads:

Outline not eligible for OLAP rules. Member &1 is a non-zero level member which is referred to by a shared member.

Note: Pay special attention to outlines with shared members, if you have any. If data values for shared members are not showing up in reports, it is possible your data does not explicitly identify the shared combination.

Frequently Asked Questions

How many “rules” can I apply to each application/database?

There can be only one rule applied to each application/database. If an application/database already has a “rule” applied, you cannot run the CRTOLAPRUL command. A message will indicate that you must delete the rule with DLTOLAPRUL and then run the CRTOLAPRUL file again.

How do I change a CRTOLAPRUL parameter or specify different sparse dimensions?

To make a change to an existing OLAP rule for an application/database, you must delete the rule with the DLTOLAPRUL command. Then, run CRTOLAPRUL and make your new specifications.

What if I restructure the outline or change dimensions?

If the outline is restructured after specifying an ISR rule with CRTOLAPRUL, the rule may become out of sync with the outline. This will not prevent the report from running. However, any performance gains from ISR will not be recognized. To help troubleshoot this problem, a warning message is placed in the Application Event Log each time the database is started. To resolve this problem, delete the existing rule with the DLTOLAPRUL command. Then rerun CRTOLAPRUL to make the rule compatible with the new outline structure.

What problems can occur and how do I solve them?

The most common ISR problem you may encounter is rejected members. To help you track and resolve this problem, an error file is created that logs the rejected members. See “Troubleshooting Intelligent Sparse Reporting” on p. 9 for more information.