# Contents

**Chapter 1 • Using Data Grids in SAS Decision Manager** .................................................. 1  
  Introduction to Data Grids .................................................. 1  
  How to Use Data Grids In SAS Decision Manager .......................... 2  
  Enable the Use of Data Grids In SAS Decision Manager ................. 3  
  Working with Rule Sets and Decisions That Reference Data Grid Terms . 4

**Chapter 2 • Data Grid Macros** .................................................................................. 7  
  Data Grid Macros Available in SAS Decision Manager .................. 7  
  Dictionary ........................................................................... 7

**Chapter 3 • Data Grid Functions** ............................................................................. 13  
  Using Data Grid Functions ...................................................... 13  
  Data Grid Functions Available in SAS Decision Manager ............ 14  
  Dictionary ........................................................................... 16
Introduction to Data Grids

Data grids are tables. In SAS Decision Manager, you can use a data grid as the value of a table cell. For example, suppose you have a table containing the data for all of the insurance policies for all of your customers. This table might look like the table shown in Table 1.1.

Table 1.1 Insurance Policy Table

<table>
<thead>
<tr>
<th>PolicyHolder</th>
<th>PolicyNumber</th>
<th>YearlyPremium</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smyth, Joe</td>
<td>453975R398</td>
<td>439.50</td>
</tr>
<tr>
<td>Smyth, Joe</td>
<td>987348P210</td>
<td>132.90</td>
</tr>
<tr>
<td>Dupree, Marcel</td>
<td>983092B228</td>
<td>334.00</td>
</tr>
<tr>
<td>Dupree, Marcel</td>
<td>274933P412</td>
<td>219.25</td>
</tr>
</tbody>
</table>
Alternatively, you can use a data grid to store the policy information as represented in Figure 1.1.

**Figure 1.1  Insurance Policy Table Using Data Grids**

<table>
<thead>
<tr>
<th>PolicyHolder</th>
<th>Policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smyth, Joe</td>
<td>PolicyNumber</td>
</tr>
<tr>
<td></td>
<td>453975R398</td>
</tr>
<tr>
<td></td>
<td>987348P210</td>
</tr>
<tr>
<td>Dupree, Marcel</td>
<td>PolicyNumber</td>
</tr>
<tr>
<td></td>
<td>983092B228</td>
</tr>
<tr>
<td></td>
<td>274933P412</td>
</tr>
</tbody>
</table>

The information in the data grid column is represented by a JavaScript Object Notation (JSON) string. For example, if the table shown in Figure 1.1 is changed to use a data grid, the table is serialized as shown in Table 1.2.

**Table 1.2  Serialized Insurance Policy Table**

<table>
<thead>
<tr>
<th>PolicyHolder</th>
<th>Policies</th>
</tr>
</thead>
<tbody>
<tr>
<td>Smyth, Joe</td>
<td>[{&quot;metadata&quot;: [{&quot;POLICYNUMBER&quot;: &quot;string&quot;}, &quot;YEARLYPREMIUM&quot;: &quot;decimal&quot;}], &quot;data&quot;: [{&quot;453975R398&quot;, 439.50}, [&quot;987348P210&quot;, 132.90]]}</td>
</tr>
<tr>
<td>Dupree, Marcel</td>
<td>[{&quot;metadata&quot;: [{&quot;POLICYNUMBER&quot;: &quot;string&quot;}, &quot;YEARLYPREMIUM&quot;: &quot;decimal&quot;}], &quot;data&quot;: [{&quot;983092B228&quot;, 324.00}, [&quot;274933P412&quot;, 219.35]]}</td>
</tr>
</tbody>
</table>

To process the data in the data grid, you could define a term named Policies of type DATAGRID and use the functions described in Chapter 3, “Data Grid Functions,” on page 13 to process the data. For example:

```sas
DATAGRID_GET(Policies, 'PolicyNumber', 2)
```

For policy holder Joe Smyth, this function call returns 987348P210.

*Note:* In this example, you define terms for only the PolicyHolder and Policies columns. You do not define terms for the PolicyNumber and YearlyPremium columns within the data grid.

---

**How to Use Data Grids In SAS Decision Manager**

*Important:* You cannot use data grids in decision condition expressions or models. If you need to use a value calculated from a data grid in a decision condition
expression, calculate the value in a rule set, assign the value to an action term, and then use the action term in the decision condition expression.

1. In SAS Management Console, enable the use of data grids by setting the brm.datagrid.type.enabled configuration property to true and verifying that the brm.runtime.codetype property is set to DS2.

2. Use the %DCM_SERIALIZEGRID macro to serialize your data grid into a JavaScript Object Notation (JSON) string. The data grid must be serialized if it is used in a rule flow or in a decision that meets either of the following criteria:
   - The rule flow or decision is used in a test.
   - The rule flow or decision is published for use in SAS Data Integration Studio.

   Note: Data grids that are used in jobs that are deployed to SAS Micro Analytic Service are automatically serialized when a job that uses the data grids is sent to the service. If your job is deployed only to SAS Micro Analytic Service, you do not need to use the %DCM_SERIALIZEGRID macro to serialize the data grids.

3. To combine data from multiple tables into one table, use the %DCM_MERGE_SERIALIZEGRID macro. This macro merges multiple data grids and scalar data together into one table.

   Note: If you write the serialized data grid term to a Base SAS table, the maximum size of the JSON string is 32,767 characters. The default size is 10,485,760 characters.


5. Define terms for the table columns that contain the data grid. Select Data Grid as the data type. See “Defining Data Grid Terms” on page 4 and “Create a Term” in SAS Decision Manager: User’s Guide for more information.

   Important: Defining terms for each column in the data grid is not necessary. Data grid columns are referred to by character literals, not by terms.

6. Use the functions described in Chapter 3, “Data Grid Functions,” on page 13 to access or modify data grid terms.

   Note: You can nest function calls. For example:

   datagrid_count(datagrid_subsetByValue(assets, 'Asset_Type', 'Savings'))

7. When you test SAS Decision Manager content that uses the data grid term, map the data grid term to the column in the input table that contains the data grid JSON string.

---

Enable the Use of Data Grids In SAS Decision Manager

The brm.datagrid.type.enabled and brm.runtime.codetype configuration properties in SAS Management Console enable the use of data grids and data grid functions in rule sets. You must have administrator privileges to enable the use of data grids.

To enable the use of data grids:

1. Complete the following steps in SAS Management Console:
a. On the Plug-ins tab, select Application Management ➤ Configuration Manager ➤ SAS Application Infrastructure ➤ Enterprise Decision Manager 3.2.


Note: If you have migrated to SAS Decision Manager 3.2, SAS Management Console displays the release number that you migrated from. For example, if you migrated from SAS Decision Manager 2.2, SAS Management Console displays Business Rules Manager 2.2.

c. On the Advanced tab, set the brm.datagrid.type.enabled property to true.

Note: Click to enable editing for a property.

d. Set the brm.runtime.codetype property to DS2.

e. Click OK to close the Business Rules Manager Web 3.2 Properties dialog box.

2. Restart SASServer7. You must restart the server in order for changes to configuration properties to take effect.

Working with Rule Sets and Decisions That Reference Data Grid Terms

Defining Data Grid Terms

Data grid terms can be imported, exported, created, edited, and added to rule sets in the same way as other terms. When you are creating or importing data grid terms, the following guidelines apply:

- Domain values are not applicable to data grid terms.
- The columns within a data grid can contain only character or numeric data.
- In the SAS Decision Manager interface, define terms for only the columns that contain the data grids. Defining terms for each column in the data grid is not necessary. Data grid columns are referred to by character literals, not by terms.

Testing and Publishing Rule Flows and Decisions

When you are testing or publishing a rule flow that uses data grids, the following guidelines apply:

- You can publish rule flows that use data grids for use by SAS Data Integration Studio, but you cannot publish them as stored processes. You can publish decisions that use data grids as Decision Logic objects for use by SAS Data Integration Studio, or you can publish them to the SAS Micro Analytic Service. For more information, see “Publish a Rule Flow” in SAS Decision Manager: User’s Guide and “Publish a Decision” in SAS Decision Manager: User’s Guide.
- Data grids and data grid functions are supported only in DS2 code. They are not supported for SAS DATA Step (DS1) code. See the description of the brm.runtime.codetype property in “Enable the Use of Data Grids In SAS Decision Manager” on page 3 for more information.
• The DS2 code generated by SAS Decision Manager can be executed in-database by the SAS Code Accelerator.

• When you are testing a rule flow or decision that uses DATAGRID terms, the DATAGRID terms must be mapped to character variables. When you are running a decision using the SAS Micro Analytic Service, the DATAGRID terms must be specified as a DATAGRID parameter.

• Data grids are not supported by SAS Real-Time Decision Manager or by risk products such as SAS Risk Dimensions and SAS Risk and Finance Workbench.

• In a decision, you cannot map DATAGRID terms to model variables.
Chapter 2
Data Grid Macros

**Data Grid Macros Available in SAS Decision Manager**

Data grids must be serialized into JavaScript Object Notation (JSON) strings before you can use the data in SAS Decision Manager. You can use the following macros to prepare grid data to be used in rules sets:

- `%DCM_SERIALIZEGRID` creates a table in which one of the columns contains a JSON string that represents a data grid.
- `%DCM_MERGE_SERIALIZEDGRIDS` merges tables that contain data grids that have been serialized with the `%DCM_SERIALIZEGRID` macro with a table that contains scalar data.

**Dictionary**

- `%DCM_SERIALIZEGRID`

  Creates a table in which one of the columns contains a JSON string that represents a data grid.
  
  **Note:** This macro trims both leading and trailing spaces from the character strings in data grid columns.

  **Syntax**

  ```
  %DCM_SERIALIZEGRID (GRIDCOLNAME=data_grid_column_name)
  ```
GRIDSOURCETABLE=table_name,
OUTPUTTABLE=results_table,
<ClassVars=class_variable1<class_variable2...>>
<GridCols=column1<column2...>)

**Required Arguments**

GRIDCOLNAME=data_grid_column_name
specifies the name for the serialized data grid column in the results table.

GRIDSOURCETABLE=table_name
specifies the table that contains the columns that are to be serialized into a data grid.

OUTPUTTABLE=results_table
specifies the name of the results table that contains the data grid column.

**Optional Arguments**

CLASSVARS=class_variable1 <class_variable2...>
specifies the class variables that control how the data is grouped into data grids. If you specify one class variable, the results table contains one data grid for each value of the class variable. If you specify more than one class variable, a separate data grid is created for each combination of values for the class variables. The class variables are written to the output results table.

GRIDCOLS=column1 <column2...>
specifies the names of the columns that are to be serialized into the data grid. If you do not specify this option, all of the columns in the table are serialized except for the columns specified by the CLASSVARS= option.

**Example: Serializing the Assets Table**

The following example serializes (creates JSON strings) the assets data for each customer in the mylib.assets table. The output table is named assetsGrid, and the data grid column in the output table is named Assets. The data in the mylib.assets table is grouped by the class variable custName, so the output table contains one row for each value of custName.

```bash
%dcm_serializeGrid(
    gridSourceTable=mylib.assets,
    gridColName=Assets,
    outputTable=assetsGrid,
    classvars=custName);
```

**Table 2.1 mylib.assets Source Table**

<table>
<thead>
<tr>
<th>CustName,</th>
<th>AssetType</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacKelly, Sara</td>
<td>Property, Primary</td>
<td>212000</td>
</tr>
<tr>
<td>MacKelly, Sara</td>
<td>Property, Investment</td>
<td>125000</td>
</tr>
<tr>
<td>Smyth, Joe</td>
<td>Property, Primary</td>
<td>234500</td>
</tr>
</tbody>
</table>
Table 2.2  assetGrid Results Table

<table>
<thead>
<tr>
<th>CustName</th>
<th>Assets</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacKelly, Sara</td>
<td>[{&quot;metadata&quot;: [{&quot;ASSETTYPE&quot;: &quot;string&quot;}, {&quot;VALUE&quot;: &quot;decimal&quot;}]], &quot;data&quot;: [{&quot;Property, Primary&quot;: 212000}, {&quot;Property, Investment&quot;: 125000}]}</td>
</tr>
<tr>
<td>Smyth, Joe</td>
<td>[{&quot;metadata&quot;: [{&quot;ASSETTYPE&quot;: &quot;string&quot;}, {&quot;VALUE&quot;: &quot;decimal&quot;}]], &quot;data&quot;: [{&quot;Property, Primary&quot;: 234500}]}</td>
</tr>
</tbody>
</table>

%DCM_MERGESERIALIZEDGRIDS

Merges tables that contain data grids that have been serialized with the %DCM_SERIALIZEGRIDS macro with a table that contains scalar data.

**Note:** This macro trims both leading and trailing spaces from the character strings in data grid columns.

**Syntax**

%DCM_MERGESERIALIZEDGRIDS ( MERGETABLE=table_name, MERGEKEY=merged_table_key, OUTPUTTABLE=results_table, GRIDTABLES=data_grid1 <data_grid2>..., GRIDMERGEKEYS=data_grid_key1 <data_grid_key2>..., GRIDCOLUMNs=data_grid_column1 <data_grid_column2>... )

**Required Arguments**

MERGETABLE=table_name

specifies the name of the table that contains the scalar data.

MERGEKEY=merge_table_key

specifies the name of the key column in the table specified by the MERGETABLE= argument.

**Note:** You can specify only one merge column key. The %DCM_MERGESERIALIZEDGRIDS macro does not support merges based on multiple key values.

OUTPUTTABLE=results_table

specifies the name of the table that contains the results of the merge.

GRIDTABLES=data_grid1 <data_grid2> ...

specifies the names of the tables that contain the data grid columns that are to be merged into the results table. The data grid columns must contain a serialized data grid produced by the %DCM_SERIALIZEGRID macro.

**Interaction**

For each data grid that is to be merged into the results table, the name of the data grid table, merge key, and data grid column must be
specified in the same order for each of the GRIDTABLES=, GRIDMERGEKEYS=, and GRIDCOLUMNS= arguments.

**GRIDMERGEKEYS=**
\[\text{data_grid_key1} < \text{data_grid_key2} > ...\]
specifies the names of the key columns in the tables specified by the GRIDTABLES= argument.

**Interaction**
The first key specified must be the key for the first table specified in the GRIDTABLES= argument, the second key must be the key for the second table, and so on.

**GRIDCOLUMNS=**
\[\text{data_grid_column1} < \text{data_grid_column2} > ...\]
specifies the names of the data grid columns in the tables specified by the GRIDTABLES= argument.

**Interaction**
The first column specified must be the name of the data grid column in the first table specified in the GRIDTABLES= argument, the second key must be the name of the data grid column in the second table, and so on.

---

**Example: Merging Debts and Assets Data Grids with Loan Request Information**

The following example merges the scalar data in the mylib.loanRequests table with the data grid columns in the tables `debtsGrid` and `assetGrid`. The key column for the scalar table and the data grid tables is `custName`. The data grid columns in the resulting output table, `mylib.loadRequestData`, are named Debts and Assets.

```
%dcm_mergeSerializedGrids(
   mergeTable=mylib.loanRequests,
   mergekey=Customer,
   outputTable=mylib.loadRequestData,
   gridTables=debtsGrid assetsGrid,
   gridMergeKeys=custName custName
   gridColumns=Debts Assets);
```

<table>
<thead>
<tr>
<th>RequestID</th>
<th>RequestedAmt</th>
<th>Customer</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME4922Mac01</td>
<td>20000</td>
<td>MacKelly, Sara</td>
</tr>
<tr>
<td>NC2W497Smy03</td>
<td>80495</td>
<td>Smyth, Joe</td>
</tr>
</tbody>
</table>

**Table 2.3**  
mylib.loanrequests Merge Table

**Table 2.4**  
`debtsGrid` Data Grid Table

<table>
<thead>
<tr>
<th>CustName</th>
<th>Debts</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacKelly, Sara</td>
<td>[{&quot;metadata&quot;: [{&quot;DEBTTYPE&quot;: &quot;string&quot;}, &quot;BALANCE&quot;: &quot;decimal&quot;}], &quot;data&quot;: [{&quot;Mortgage&quot;, 80053.16}, [&quot;CreditCard&quot;, 2143.68]}]</td>
</tr>
</tbody>
</table>
**Table 2.5** assetsGrid Data Grid Table

<table>
<thead>
<tr>
<th>CustName</th>
<th>Assets</th>
</tr>
</thead>
<tbody>
<tr>
<td>MacKelly, Sara</td>
<td>[&quot;metadata&quot;: [&quot;ASSETTYPE&quot;: &quot;string&quot;], &quot;VALUE&quot;: &quot;decimal&quot;], &quot;data&quot;: [&quot;Property, Primary&quot;, 212000], [&quot;Property, Investment&quot;, 125000]]</td>
</tr>
<tr>
<td>Smyth, Joe</td>
<td>[&quot;metadata&quot;: [&quot;ASSETTYPE&quot;: &quot;string&quot;], &quot;VALUE&quot;: &quot;decimal&quot;], &quot;data&quot;: [&quot;Property, Primary&quot;, 234500]]</td>
</tr>
</tbody>
</table>

**Table 2.6** mylib.loanRequestData Results Table

<table>
<thead>
<tr>
<th>RequestID</th>
<th>RequestAmt</th>
<th>Customer</th>
<th>Debts</th>
<th>Assets</th>
</tr>
</thead>
<tbody>
<tr>
<td>ME4922Mac01</td>
<td>20000</td>
<td>MacKelly, Sara</td>
<td>[&quot;metadata&quot;: [&quot;DEBTTYPE&quot;: &quot;string&quot;], &quot;BALANCE&quot;: &quot;decimal&quot;], &quot;data&quot;: [&quot;Mortgage&quot;, 80053.16], [&quot;CreditCard&quot;, 2143.68]]</td>
<td>[&quot;metadata&quot;: [&quot;ASSETTYPE&quot;: &quot;string&quot;], &quot;VALUE&quot;: &quot;decimal&quot;], &quot;data&quot;: [&quot;Property, Primary&quot;, 212000], [&quot;Property, Investment&quot;, 125000]]</td>
</tr>
<tr>
<td>NC2497Smy03</td>
<td>80495</td>
<td>Smyth, Joe</td>
<td>.</td>
<td>[&quot;metadata&quot;: [&quot;ASSETTYPE&quot;: &quot;string&quot;], &quot;VALUE&quot;: &quot;decimal&quot;], &quot;data&quot;: [&quot;Property, Primary&quot;, 234500]]</td>
</tr>
</tbody>
</table>
Chapter 3
Data Grid Functions

Using Data Grid Functions

You must use SAS Decision Manager data grid functions to process data grid terms. SAS Decision Manager supplies several functions for use with data grids.
Comparison Operators

You can use any of the following operators with functions that compare values such as the DATAGRID_FILTEREDGET, DATAGRID_FILTEREDSETALL, and DATAGRID_MATCHCOUNT functions.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EQ, =, ==</td>
<td>Equals</td>
</tr>
<tr>
<td>NE, !=, ^=, &lt;&gt;</td>
<td>Not equal to</td>
</tr>
<tr>
<td>GT, &gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>LT, &lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>LE, &lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>GE, &gt;=</td>
<td>Greater than or equal to</td>
</tr>
</tbody>
</table>

Enable Logging for Data Grid Functions

You can enable logging for data grid functions by adding the DCM_WSLogAppender definition to the logconfig.xml file on your workspace server. To enable logging, add the following code to the file SAS-configuration-directory\Lev1\SASApp\WorkspaceServer\logconfig.xml. Add this code after the WSLogAppender definition and before the immutable logger definitions.

```xml
<!-- Write log messages to workspace server sas log -->
<appender name="DCM_WSLogAppender" class="sLogAppender">
  <param name="LogName" value="WSLOG"/>
  <layout>
    <param name="ConversionPattern" value="%-5p:  %m"/>
  </layout>
</appender>

<logger name="App.SASDCM">
  <level value="Info"/>
  <appender-ref ref="DCM_WSLogAppender"/>
</logger>
```

Data Grid Functions Available in SAS Decision Manager

Functions are categorized by the types of values that they return or type of operation they perform. Each function belongs to one of the following categories:
Create, Update, or Delete
Create, copy, or clear the data in a data grid.

Information
Return information about an entire data grid or about a column or row in the data grid.

Join or Append
Join or append two data grids.

Retrieve Values
Retrieve values from a data grid.

Set Values
Set the value of cells in a data grid.

Statistical
Perform statistical calculations on values in the data grid.

Subset and Sort
Subset or sort a data grid.

The following table provides brief descriptions of the data grid functions.

<table>
<thead>
<tr>
<th>Category</th>
<th>Language Elements</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Create, Update, or Delete</td>
<td>DATAGRID_COPY (p. 16)</td>
<td>Returns a copy of the specified data grid, or returns an empty data grid if the operation is unsuccessful.</td>
</tr>
<tr>
<td>Information</td>
<td>DATAGRID_COUNT (p. 17)</td>
<td>Returns the number of rows in the specified data grid, or returns a negative number if an error occurs.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_DISTRIBUTEROWCOUNT (p. 17)</td>
<td>Returns the number of unique rows in the specified data grid, or returns a negative number if an error occurs.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_GRIDMATCHCOUNT (p. 22)</td>
<td>Returns the number of rows for which the value in the specified column in one data grid matches the value in the specified column in another data grid, or returns a negative number if an error occurs.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_MATCHCOUNT (p. 25)</td>
<td>Returns the number of rows in the specified column for which the specified comparison evaluates to true, or returns a negative number if an error occurs.</td>
</tr>
<tr>
<td>Join or Append</td>
<td>DATAGRID_INNERJOIN (p. 22)</td>
<td>Performs an inner join of two data grids and returns the resulting data grid, or returns an empty data grid if the operation is unsuccessful.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_LEFTJOIN (p. 23)</td>
<td>Performs a left join of two data grids and returns the resulting data grid, or returns an empty data grid if the operation is unsuccessful.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_RIGHTJOIN (p. 27)</td>
<td>Performs a right join of two data grids and returns the resulting data grid, or returns an empty data grid if the operation is unsuccessful.</td>
</tr>
<tr>
<td>Retrieve Values</td>
<td>DATAGRID_FILTEREDGET (p. 18)</td>
<td>Returns the value in the first row in the specified column for which the specified comparison evaluates to true. If the operation is unsuccessful, the function returns an empty string for character values or a missing value (.) for numeric values.</td>
</tr>
<tr>
<td>Category</td>
<td>Language Elements</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Create, Update, or Delete</td>
<td>DATAGRID_GET</td>
<td>Returns the value of the cell in the specified row and column. If the operation is unsuccessful, the function returns an empty string for character values or a missing value (.) for numeric values.</td>
</tr>
<tr>
<td>Set Values</td>
<td>DATAGRID_FILTEREDSET (p. 19)</td>
<td>Sets the value in the specified row and column to the specified value if the specified comparison evaluates to true. The function returns 1 if the operation is successful and a negative number if the operation is unsuccessful.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_FILTEREDSETALL (p. 20)</td>
<td>Sets the cell in the specified column to the specified value for each row for which the specified comparison evaluates to true. The function returns the number of cells that were set to the specified value, or it returns a negative number if the operation is unsuccessful.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_SET</td>
<td>Assigns the specified value to the specified row and column. The function returns 1 if the operation is successful, or returns a negative number if it is unsuccessful.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_SETALL</td>
<td>Assigns the specified value to all rows in the specified column. The function returns 1 if the operation is successful, or returns a negative number if it is unsuccessful.</td>
</tr>
<tr>
<td>Statistical</td>
<td>DATAGRID_MAX</td>
<td>Returns the maximum value for the specified column, or returns a missing value (.) if an error occurs.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_MEAN</td>
<td>Returns the mean value for the specified column, or returns a missing value (.) if an error occurs.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_MEDIAN</td>
<td>Returns the median value for the specified column, or returns a missing value (.) if an error occurs.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_MIN</td>
<td>Returns the minimum value that appears in the specified column, or returns a missing value (.) if an error occurs.</td>
</tr>
<tr>
<td></td>
<td>DATAGRID_SUM</td>
<td>Returns the sum of the values in the specified column, or returns a missing value (.) if an error occurs.</td>
</tr>
<tr>
<td>Subset and Sort</td>
<td>DATAGRID_SUBSETBYVALUE (p. 29)</td>
<td>Returns a data grid that contains the rows for which the specified comparison evaluates to true, or returns an empty data grid if the operation is unsuccessful.</td>
</tr>
</tbody>
</table>

**Dictionary**

**DATAGRID_COPY**

Returns a copy of the specified data grid, or returns an empty data grid if the operation is unsuccessful.

*Category:* Create, Update, or Delete
**DATAGRID_COPY**

Syntax

`DATAGRID_COPY(dataGrid)`

*Required Argument*

`dataGrid` specifies the name of the data grid. This argument must be a term of type `DATAGRID`.

---

**DATAGRID_COUNT**

Returns the number of rows in the specified data grid, or returns a negative number if an error occurs.

*Category:* Information

*Returned data type:* `INTEGER`

**Syntax**

`DATAGRID_COUNT(dataGrid)`

*Required Argument*

`dataGrid` specifies the name of the data grid. This argument must be a term of type `DATAGRID`.

---

**DATAGRID_DISTINCTROWCOUNT**

Returns the number of unique rows in the specified data grid, or returns a negative number if an error occurs.

*Category:* Information

*Returned data type:* `INTEGER`

**Syntax**

`DATAGRID_DISTINCTROWCOUNT(dataGrid)`

*Required Argument*

`dataGrid` specifies the name of the data grid. This argument must be a term of type `DATAGRID`. 
DATAGRID_FILTEREDGET

Returns the value in the first row in the specified column for which the specified comparison evaluates to true. If the operation is unsuccessful, the function returns an empty string for character values or a missing value (.) for numeric values.

**Category:** Retrieve Values

**Returned data type:** STRING, DOUBLE

**Tip:** This function returns a value for only the first row in which the comparison evaluates to true. To find all rows for which a specified comparison evaluates to true, use the “DATAGRID_SUBSETBYVALUE” function.

**Syntax**

\[
\text{DATAGRID\_FILTEREDGET} (\text{dataGrid, columnReturned, filterColumn, operator, termOrValue})
\]

**Required Arguments**

\textit{dataGrid}

specifies the name of the data grid. This argument must be a term of type DATAGRID.

\textit{columnReturned}

specifies the name of the column whose value you want to know. You can specify a column name in single quotation marks or a term that evaluates to a column name.

\textit{filterColumn}

specifies the name of the column whose value is to be compared to \textit{termOrValue}. You can specify a column name in single quotation marks or a term that evaluates to a column name.

\textit{operator}

specifies one of the following operators shown in “Comparison Operators” on page 14. You can specify the name of a character variable that evaluates to one of these operators, or you can specify the operator enclosed in single quotation marks.

\textit{termOrValue}

specifies the value to compare to the value of \textit{filterColumn}. You can specify a number, a character string enclosed in single quotation marks, the name of a term, or an expression. The value that you specify must be or must evaluate to the same data type as \textit{filterColumn}.

**Details**

The DATAGRID_FILTEREDGET function compares the value in \textit{filterColumn} to the specified \textit{termOrValue} by using the comparison operator. The comparison is as follows:

\[
\text{filterColumn} \ \text{operator} \ \text{termOrValue}
\]

For the first row for which the comparison evaluates to true, this function returns the value of \textit{columnReturned}.
Example

The following example determines whether the value of the riskGroup column is equal to Low, and if so, returns the value of the approvalStatus column:

\[\text{DATAGRID\_FILTEREDGET(DebtEval,'approvalStatus','riskGroup','EQ','Low')}\]

**DATAGRID\_FILTEREDSET**

Sets the value in the specified row and column to the specified value if the specified comparison evaluates to true. The function returns 1 if the operation is successful and a negative number if the operation is unsuccessful.

*Category:* Set Values  
*Returned data type:* INTEGER

**Syntax**

\[\text{DATAGRID\_FILTEREDSET}(dataGrid, columnToAssign, rowNumber, filterColumn, operator, termOrValue, valueToAssign)\]

**Required Arguments**

*dataGrid*  
specifies the name of the data grid. This argument must be a term of type DATAGRID.

*columnToAssign*  
specifies the name of the column to be assigned the specified value. You can specify a column name in single quotation marks or a term that evaluates to a column name.

*rowNumber*  
specifies the row number that contains the cell to be assigned the specified value. You can specify a number or a term that evaluates to a number.

*filterColumn*  
specifies the name of the column whose value is to be compared to termOrValue. You can specify a column name in single quotation marks or a term that evaluates to a column name.

*operator*  
specifies one of the following operators shown in “Comparison Operators” on page 14. You can specify the name of a character variable that evaluates to one of these operators, or you can specify the operator enclosed in single quotation marks.

*termOrValue*  
specifies the value to compare to the value of filterColumn. You can specify a number, a character string enclosed in single quotation marks, the name of a term, or an expression. The value that you specify must be or must evaluate to the same data type as filterColumn.

*valueToAssign*  
specifies the value to assign to the cell at rowNumber, columnToAssign. You can specify a number, a character value enclosed in single quotation marks, or the name of a term.
Details

The DATAGRID_FILTEREDSET function compares the value of filterColumn to the value of termOrValue using the comparison operator. The comparison is as follows:

\[ \text{filterColumn} \ \text{operator} \ \text{termOrValue} \]

If the comparison evaluates to true, this function sets the value of rowNumber, columnToAssign to the value specified by valueToAssign.

Example

For the data grid row specified by the value of the customer variable, the DATAGRID_FILTEREDSET function sets the column riskGroup to High if the value of the Debts column is greater than the value of assets term:

\[ \text{DATAGRID_FILTEREDSET(DebtEval,'riskGroup',customer,'debts','GT',assets,'High')} \]

### DATAGRID_FILTEREDSETALL

Sets the cell in the specified column to the specified value for each row for which the specified comparison evaluates to true. The function returns the number of cells that were set to the specified value, or it returns a negative number if the operation is unsuccessful.

**Category:** Set Values

**Returned data type:** INTEGER

**Note:** This function returns the number of cells that were set to the specified value.

### Syntax

\[
\text{DATAGRID_FILTEREDSETALL} \ (\text{dataGrid, columnToAssign, filterColumn, operator, termOrValue, valueToAssign})
\]

**Required Arguments**

- **dataGrid**
  - specifies the name of the data grid. This argument must be a term of type DATAGRID.

- **columnToAssign**
  - specifies the name of the column to be assigned the specified value. You can specify a column name in single quotation marks or a term that evaluates to a column name.

- **filterColumn**
  - specifies the name of the column whose value is to be compared to termOrValue. You can specify a column name in single quotation marks or a term that evaluates to a column name.

- **operator**
  - specifies one of the following operators shown in “Comparison Operators” on page 14. You can specify the name of a character variable that evaluates to one of these operators, or you can specify the operator enclosed in single quotation marks.

- **termOrValue**
  - specifies the value to compare. This can be a term or a literal value enclosed in single quotation marks.
termOrValue
  specifies the value to compare to the value of filterColumn. You can specify a number, a character string enclosed in single quotation marks, the name of a term, or an expression. The value that you specify must be or must evaluate to the same data type as filterColumn.

valueToAssign
  specifies the value to assign to the appropriate cells in columnToAssign. You can specify a number, a character value enclosed in single quotation marks, the name of a term, or an expression. The value that you specify must be or must evaluate to the same data type as columnToAssign.

Details

For each row in the data grid, the DATAGRID_FILTEREDSETALL function compares the value of filterColumn to the value of termOrValue by using the comparison operator. The comparison is as follows:

   filterColumn operator termOrValue

If the comparison evaluates to true, this function sets the value of the cell in column columnToAssign to the value specified by valueToAssign.

Example

This example sets the value of the cell in column riskGroup to the value of the highGroup term for all rows in the data grid DebtEval for which the value of the Debts column is greater than the value of the assets term:

   DATAGRID_FILTEREDSETALL(DebtEval,'riskGroup','debts','GT',assets,highGroup)

---

**DATAGRID_GET**

Returns the value of the cell in the specified row and column. If the operation is unsuccessful, the function returns an empty string for character values or a missing value (.) for numeric values.

<table>
<thead>
<tr>
<th>Category:</th>
<th>Retrieve Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned data type:</td>
<td>STRING, DOUBLE</td>
</tr>
</tbody>
</table>

**Syntax**

   DATAGRID_GET (dataGrid, column, rowNumber)

**Required Arguments**

- **dataGrid**
  specifies the name of the data grid. This argument must be a term of type DATAGRID.

- **column**
  specifies the name of the column in the data grid. You can specify a column name in single quotation marks or a term that evaluates to a column name.
**rowNumber**

specifies the row number in the data grid. You can specify a number or a term that evaluates to a number.

---

**DATAGRID_GRIDMATCHCOUNT**

Returns the number of rows for which the value in the specified column in one data grid matches the value in the specified column in another data grid, or returns a negative number if an error occurs.

<table>
<thead>
<tr>
<th>Category:</th>
<th>Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned data type:</td>
<td>INTEGER</td>
</tr>
</tbody>
</table>

**Syntax**

```plaintext```
DATAGRID_GRIDMATCHCOUNT (dataGrid1, dataGrid2, column1, column2)
```plaintext```

**Required Arguments**

- `dataGrid1`
- `dataGrid2`
- `column1`
- `column2`

**dataGrid1**
specifies the names of data grids. These arguments must be terms of type DATAGRID.

**column1**
specifies the name of the column in `dataGrid1` that you want to compare to a column in `dataGrid2`. You can specify a column name in single quotation marks or a term that evaluates to a column name.

**column2**
specifies the name of the column in `dataGrid2`. You can specify a column name in single quotation marks or a term that evaluates to a column name.

**Details**

The DATAGRID_GRIDMATCHCOUNT function returns the number of rows in which the value of `dataGrid1, column1` match the value of `dataGrid2, column2`.

---

**DATAGRID_INNERJOIN**

Performs an inner join of two data grids and returns the resulting data grid, or returns an empty data grid if the operation is unsuccessful.

<table>
<thead>
<tr>
<th>Category:</th>
<th>Join or Append</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned data type:</td>
<td>DATAGRID</td>
</tr>
</tbody>
</table>

**Syntax**

```plaintext```
DATAGRID_INNERJOIN (dataGrid1, dataGrid2, keyColumn1, keyColumn2)
```plaintext```

**DATAGRID_INNERJOIN**

Performs an inner join of two data grids and returns the resulting data grid, or returns an empty data grid if the operation is unsuccessful.

<table>
<thead>
<tr>
<th>Category:</th>
<th>Join or Append</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned data type:</td>
<td>DATAGRID</td>
</tr>
</tbody>
</table>

**Syntax**

```plaintext```
DATAGRID_INNERJOIN (dataGrid1, dataGrid2, keyColumn1, keyColumn2)
```plaintext```
**Required Arguments**

- `dataGrid1`
- `dataGrid2`
  
  specifies the names of the two data grids to be joined.

- `keyColumn1`
  
  specifies the name of the key column in `dataGrid1`. You can specify a column name in single quotation marks or a term that evaluates to a column name.

- `keyColumn2`
  
  specifies the name of the key column in `dataGrid2`. You can specify a column name in single quotation marks or a term that evaluates to a column name.

**Details**

The DATAGRID_INNERJOIN function performs an inner join of two data grids for which `dataGrid1.keyColumn1` equals `dataGrid2.keyColumn2` and returns a data grid that contains the results of the join.

When the column names in the two data grids are identical, the columns from the left side of the join are added to the resulting data grid.

---

**DATAGRID_LEFTJOIN**

Performs a left join of two data grids and returns the resulting data grid, or returns an empty data grid if the operation is unsuccessful.

- **Category:** Join or Append
- **Returned data type:** DATAGRID

**Syntax**

`DATAGRID_LEFTJOIN(dataGrid1, dataGrid2, keyColumn1, keyColumn2)`

**Required Arguments**

- `dataGrid1`
- `dataGrid2`
  
  specifies the names of the two data grids to be joined.

- `keyColumn1`
  
  specifies the name of the key column in `dataGrid1`. You can specify a column name in single quotation marks or a term that evaluates to a column name.

- `keyColumn2`
  
  specifies the name of the key column in `dataGrid2`. You can specify a column name in single quotation marks or a term that evaluates to a column name.

**Details**

The DATAGRID_LEFTJOIN function performs a left join of two data grids for which `dataGrid1.keyColumn1` equals `dataGrid2.keyColumn2` and returns a data grid that contains the results of the join.
When the column names in the two data grids are identical, the columns from the left side of the join are added to the resulting data grid.

**DATAGRID_LOOKUPMATCHCOUNT**

Returns the number of rows for which the value in the specified column matches a lookup key in the specified lookup table, or returns a negative number if an error occurs.

- **Valid in:** Condition and action expressions
- **Returned data type:** INTEGER

**Syntax**

```
DATAGRID_LOOKUPMATCHCOUNT('lookupTable', dataGrid, column)
```

**Required Arguments**

- `'lookupTable'`
  - specifies the name of the lookup table.

- `dataGrid`
  - specifies the name of the data grid. This argument must be a term of type DATAGRID.

- `column`
  - specifies the name of the column in the data grid whose value is compared to the key values in the lookup table. You can specify a column name in single quotation marks or a term that evaluates to a column name.

**DATAGRID_LOOKUPSUBSET**

Returns a data grid that contains the rows for which the value in the specified column matches a lookup key in the specified lookup table, or returns an empty data grid if the operation is unsuccessful.

- **Valid in:** Action expressions only
- **Returned data type:** DATAGRID

**Syntax**

```
DATAGRID_LOOKUPSUBSET('lookupTable', dataGrid, column)
```

**Required Arguments**

- `'lookupTable'`
  - specifies the name of the lookup table.

- `dataGrid`
  - specifies the name of the data grid. This argument must be a term of type DATAGRID.
column
specifies the name of the column in the data grid whose values are compared to the key values in the lookup table. You can specify a column name in single quotation marks or a term that evaluates to a column name.

### DATAGRID_MATCHCOUNT

Returns the number of rows in the specified column for which the specified comparison evaluates to true, or returns a negative number if an error occurs.

**Category:** Information  
**Returned data type:** INTEGER

#### Syntax

**DATAGRID_MATCHCOUNT** \( \text{(dataGrid, filterColumn, operator, termOrValue)} \)

#### Required Arguments

- **dataGrid**
  specifies the name of the data grid. This argument must be a term of type DATAGRID.

- **filterColumn**
  specifies the name of the column whose value is to be compared to termOrValue. You can specify a column name in single quotation marks or a term that evaluates to a column name.

- **operator**
  specifies one of the following operators shown in “Comparison Operators” on page 14. You can specify the name of a character variable that evaluates to one of these operators, or you can specify the operator enclosed in single quotation marks.

- **termOrValue**
  specifies the value to compare to the value of filterColumn. You can specify a number, a character string enclosed in single quotation marks, the name of a term, or an expression. The value that you specify must be or must evaluate to the same data type as filterColumn.

### DATAGRID_MAX

Returns the maximum value for the specified column, or returns a missing value (.) if an error occurs.

**Category:** Statistical  
**Returned data type:** DOUBLE

#### Syntax

**DATAGRID_MAX** \( \text{(dataGrid, column)} \)
**Required Arguments**

*dataGrid*

specifies the name of the data grid. This argument must be a term of type DATAGRID.

*column*

specifies the name of the column in the data grid. You can specify a column name in single quotation marks or a term that evaluates to a column name.

---

**DATAGRID_MEAN**

Returns the mean value for the specified column, or returns a missing value (.) if an error occurs.

<table>
<thead>
<tr>
<th>Category</th>
<th>Statistical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned data type</td>
<td>DOUBLE</td>
</tr>
</tbody>
</table>

**Syntax**

DATAGRID_MEAN (dataGrid, column)

**Required Arguments**

*dataGrid*

specifies the name of the data grid. This argument must be a term of type DATAGRID.

*column*

specifies the name of the column in the data grid. You can specify a column name in single quotation marks or a term that evaluates to a column name.

---

**DATAGRID_MEDIAN**

Returns the median value for the specified column, or returns a missing value (.) if an error occurs.

<table>
<thead>
<tr>
<th>Category</th>
<th>Statistical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned data type</td>
<td>DOUBLE</td>
</tr>
</tbody>
</table>

**Syntax**

DATAGRID_MEDIAN (dataGrid, column)

**Required Arguments**

*dataGrid*

specifies the name of the data grid. This argument must be a term of type DATAGRID.
column
specifies the name of the column in the data grid. You can specify a column name in single quotation marks or a term that evaluates to a column name.

---

**DATAGRID_MIN**

Returns the minimum value that appears in the specified column, or returns a missing value (.) if an error occurs.

<table>
<thead>
<tr>
<th>Category:</th>
<th>Statistical</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned data type:</td>
<td>DOUBLE</td>
</tr>
</tbody>
</table>

### Syntax

**DATAGRID_MIN** *(dataGrid, column)*

### Required Arguments

**dataGrid**

specifies the name of the data grid. This argument must be a term of type DATAGRID.

**column**

specifies the name of the column in the data grid. You can specify a column name in single quotation marks or a term that evaluates to a column name.

---

**DATAGRID_RIGHTJOIN**

Performs a right join of two data grids and returns the resulting data grid, or returns an empty data grid if the operation is unsuccessful.

<table>
<thead>
<tr>
<th>Category:</th>
<th>Join or Append</th>
</tr>
</thead>
<tbody>
<tr>
<td>Returned data type:</td>
<td>DATAGRID</td>
</tr>
</tbody>
</table>

### Syntax

**DATAGRID_RIGHTJOIN** *(dataGrid1, dataGrid2, keyColumn1, keyColumn2)*

### Required Arguments

**dataGrid1**

**dataGrid2**

specifies the names of the two data grids to be joined.

**keyColumn1**

specifies the name of the key column in *dataGrid1*. You can specify a column name in single quotation marks or a term that evaluates to a column name.
keyColumn2
specifies the name of the key column in dataGrid2. You can specify a column name
in single quotation marks or a term that evaluates to a column name.

Details
The DATAGRID_RIGHTJOIN function performs a right join of two data grids for
which dataGrid1.keyColumn1 equals dataGrid2.keyColumn2. It returns a data grid that
contains the results.

When the column names in the two data grids are identical, the columns from the right
side of the join are added to the resulting data grid.

DATAGRID_SET
Assigns the specified value to the specified row and column. The function returns 1 if the operation is
successful, or returns a negative number if it is unsuccessful.

Category: Set Values
Returned data type: INTEGER

Syntax
DATAGRID_SET (dataGrid, column, rowNumber, termOrValue)

Required Arguments
dataGrid
specifies the name of the data grid. This argument must be a term of type
DATAGRID.

column
specifies the name of the column whose value is to be set. You can specify a column
name in single quotation marks or a term that evaluates to a column name.

rowNumber
specifies the number of the row to be set. You can specify a number or a term that
evaluates to a number.

termOrValue
specifies the value to be assigned to column. You can specify a number, a character
value enclosed in single quotation marks, the name of a term, or an expression. The
value that you specify must be or must evaluate to the same data type as column.

DATAGRID_SETALL
Assigns the specified value to all rows in the specified column. The function returns 1 if the operation is
successful, or returns a negative number if it is unsuccessful.

Category: Set Values
Returned data type: INTEGER
Syntax

**DATAGRID_SETALL** (dataGrid, column, termOrValue)

**Required Arguments**

*dataGrid*

specifies the name of the data grid. This argument must be a term of type DATAGRID.

*column*

specifies the name of the column whose values are to be set.

*termOrValue*

specifies the value to be assigned to *column*. You can specify a number, a character value enclosed in single quotation marks, the name of a term, or an expression. The value that you specify must be or must evaluate to the same data type as *column*.

---

**DATAGRID_SUBSETBYVALUE**

Returns a data grid that contains the rows for which the specified comparison evaluates to true, or returns an empty data grid if the operation is unsuccessful.

**Category:** Subset and Sort

**Returned data type:** DATAGRID

Syntax

**DATAGRID_SUBSETBYVALUE** (dataGrid, filterColumn, operator, termOrValue)

**Required Arguments**

*dataGrid*

specifies the name of the data grid. This argument must be a term of type DATAGRID.

*filterColumn*

specifies the name of the column whose value is to be compared to *termOrValue*. You can specify a column name in single quotation marks or a term that evaluates to a column name.

*operator*

specifies one of the following operators shown in “Comparison Operators” on page 14. You can specify the name of a character variable that evaluates to one of these operators, or you can specify the operator enclosed in single quotation marks.

*termOrValue*

specifies the value to compare to the value of *filterColumn*. You can specify a number, a character string enclosed in single quotation marks, the name of a term, or an expression. The value that you specify must be or must evaluate to the same data type as *filterColumn*. 
Details
The DATAGRID_SUBSETBYVALUE function compares the value of filterColumn to termOrValue by using the comparison operator. The comparison is as follows:

filterColumn operator termOrValue

The function returns a data grid that contains all of the rows for which the comparison evaluates to true.

DATAGRID_SUM
Returns the sum of the values in the specified column, or returns a missing value (.) if an error occurs.

**Category:** Statistical
**Returned data type:** DOUBLE

**Syntax**

DATAGRID_SUM (dataGrid, column)

**Required Arguments**

*dataGrid*
- specifies the name of the data grid. This argument must be a term of type DATAGRID.

*column*
- specifies the name of the column in the data grid. You can specify a column name in single quotation marks or a term that evaluates to a column name.
Gain Greater Insight into Your SAS® Software with SAS Books.

Discover all that you need on your journey to knowledge and empowerment.

support.sas.com/bookstore for additional books and resources.