



DASHBOARD BUILDER™ FOR MICROSOFT® ACCESS™

Application Guide

Version 1.5

04.07.2008

This document is copyright © 2007-2008 OpenGate Software. The information contained in this document is subject to change without notice. If you find any problems in the documentation, please report them to us in writing. This document is not warranted to be error-free. Except as may be expressly permitted in your license agreement for these Programs, no part of these Programs may be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose.

Dashboard Builder is a trademark of OpenGate Software Inc.

Microsoft and the Office logo are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries.



TABLE OF CONTENTS

1	GENERAL CONCEPTS.....	3
2	DEPLOYMENT OPTIONS.....	3
2.1	USING DASHBOARD BUILDER AS A STAND-ALONE APPLICATION	3
2.2	USING DASHBOARD BUILDER WITHIN AN EXISTING DATABASE	4
2.3	USING DASHBOARD BUILDER WITH MICROSOFT® SQL SERVER™	4
2.4	USING DASHBOARD BUILDER WITH MYSQL® SERVER	5
3	MANAGING DASHBOARDS.....	5
3.1	CREATING A DASHBOARD	5
3.1.1	<i>Dashboard Layouts.....</i>	5
3.1.2	<i>Managing Groups.....</i>	6
3.1.3	<i>Assigning Metrics.....</i>	6
3.1.4	<i>Dashboard Filters.....</i>	6
3.2	CREATING METRICS	8
3.2.1	<i>Basic Metric Setup.....</i>	9
3.2.2	<i>Advanced Settings.....</i>	11
3.3	LAUNCHING DASHBOARDS FROM MACROS OR CODE	12
4	ADVANCED SETUP	12
4.1	LAUNCH A DASHBOARD ON STARTUP	12
4.2	CONFIGURING THE LOG	13
5	ADVANCED CUSTOMIZATION	14
5.1	ADDING METERS TO YOUR OWN FORMS	14
5.1.1	<i>Create the Meter Group.....</i>	14
5.1.2	<i>Modify the form_load event.....</i>	15
5.1.3	<i>Enable drilldown and mouse-over effects</i>	15
5.2	OBTAINING A METRIC VALUE FROM CODE	16
6	UPGRADING DASHBOARD BUILDER	16
6.1	OPTION 1: IMPORT YOUR DASHBOARD AND METRIC DEFINITION TABLES INTO THE NEWEST VERSION OF DASHBOARD BUILDER.	17
6.2	OPTION 2: IMPORT ALL OF THE DASHBOARD BUILDER OBJECTS INTO YOUR EXISTING DATABASE. 17	
	RELEASE HISTORY	18



1 General Concepts

Dashboard Builder™ allows users to create an unlimited number of dashboards and business metrics. A conceptual model of how Dashboard Builder operates is outlined in the following diagram.

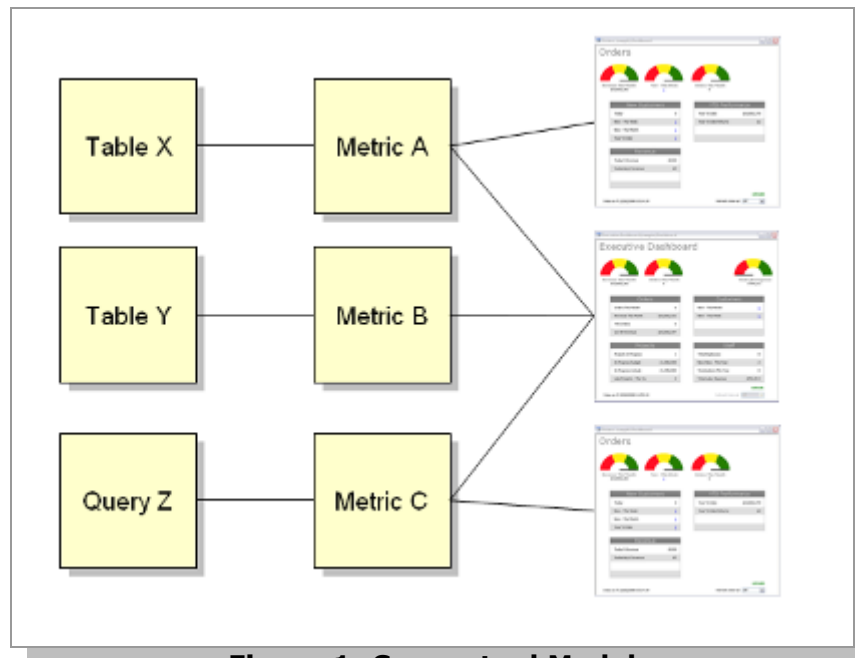


Figure 1: Conceptual Model

As shown above, note that each metric is associated with a single data source. That data source may be a table or query. Each metric can be associated with one or more dashboards, as shown at right.

2 Deployment Options

2.1 USING DASHBOARD BUILDER AS A STAND-ALONE APPLICATION

As a stand-alone database, you will need to link tables from other databases to your Dashboard Builder database. You may link tables from one or more different databases, which may be other Microsoft Access, Microsoft SQL Server, or MySQL Server databases.

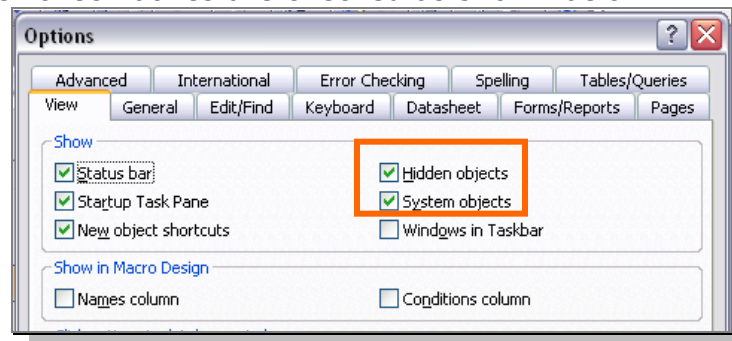
To link tables from another Microsoft Access database, open the Dashboard Administration form by clicking "Manage" from the welcome screen. Then select the "Link Tables" button from the bottom of the form.



2.2 USING DASHBOARD BUILDER WITHIN AN EXISTING DATABASE

To import Dashboard Builder into your existing database file, follow these steps:

1. We recommend you first backup your current application. You can backup your database when it is open by selecting **File>>Backup**, or by copying and pasting your database file in Windows Explorer.
2. Select **Tools>>Options** from the File menu.
3. In the Options dialog, select the View tab.
4. At the top right, make certain the "Hidden objects" and "System objects" check boxes are checked as shown below.



5. Select **OK**.
6. Select **File>>Get External Data>>Import...** from the File menu.
7. Select the Dashboard Builder file you have downloaded or received in the File Open dialog.
8. In the Import Objects dialog select all tables except in the Tables tab except those that begin with "Msys." You can also exclude the sample tables: **tblCustomer, tblEmployee, tblOrder, tblProject**
9. Select each remaining tab, and for each, click on the "Select All" button.
10. Click **OK**.

Important!

Any objects with the same name will be imported as "Name1." That is, Access will append a number to the end of the duplicate-named object.

2.3 USING DASHBOARD BUILDER WITH MICROSOFT® SQL SERVER™

To use Dashboard Builder with Microsoft SQL Server, you will need to establish an ODBC database connection to your desired database and create links to the desired tables on the SQL Server. OpenGate Software recommends you follow the instructions provided by Microsoft to link SQL Server tables to an Access database. The instructions can be found here:

**Access 2000-2003**

<http://office.microsoft.com/en-us/access/HP052730091033.aspx>

Access 2007

<http://office.microsoft.com/en-us/access/HP052730091033.aspx>

2.4 USING DASHBOARD BUILDER WITH MYSQL® SERVER

To use Dashboard Builder with MySQL Server, you will need to establish an ODBC database connection to your desired database and create links to the desired tables on the MySQL Server. OpenGate Software recommends you follow the instructions provided by MySQL to link MySQL Server tables to an Access database. The instructions can be found here:

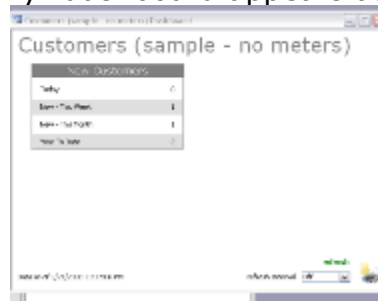
<http://dev.mysql.com/tech-resources/articles/migrating-from-microsoft.html>

3 Managing Dashboards**3.1 CREATING A DASHBOARD**

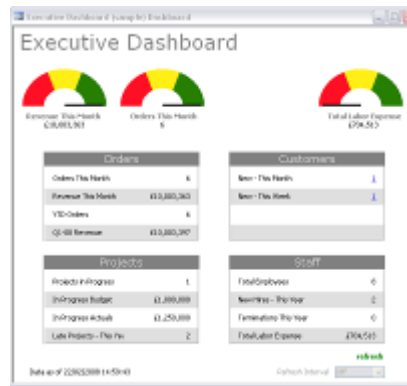
To create a new dashboard, simply select the “New Dashboard” button from the Dashboard Administration screen. Supply a name for your dashboard, and you are ready to begin assigning metrics to the dashboard.

3.1.1 DASHBOARD LAYOUTS

Dashboard Builder supports two layouts: “Data Only” and “Data and Meters.” The “Data Only” dashboard appears as follows:



Whereas the “Data and Meters” dashboard will appear as follows:



3.1.2 MANAGING GROUPS

All Dashboard Builder dashboards support up to four groups of separate metrics. A dashboard group is only displayed if you provide a name for the group. You can change the name of any group by simply selecting the text and typing your desired name. To hide a group, simply click the red 'x' beside the name. When a group has no name, it will not be displayed on the dashboard.

3.1.3 ASSIGNING METRICS

To assign a metric, simply select the metric name from the dropdown box. There are no restrictions to which metrics you assign to which groups. For example, you may choose to assign metrics associated with four different tables to the four metric spaces contained within one group. Note that only metrics that have been validated will appear in the dropdown selection boxes on screen. Refer to section 3.2 for more information about creating metrics.

When assigning metrics for the dashboard meters, note that only those metrics with a defined "Green Limit" and "Red Limit" will appear in the dropdown list. Refer to section 3.2.2 of this document for information about creating metrics for meter dashboards.

3.1.4 DASHBOARD FILTERS

In some cases, you may want your users to refine their data selection at the time they launch a dashboard. For example, where you have multiple offices or facilities in your database, and do not want to create separate dashboards and metrics for each facility. In this case, you can configure Dashboard Builder to prompt the user to choose a specific filter value from a pre-defined list prior to launching the dashboard.

3.1.4.1 Creating Filter Values

Filter values are stored in the table tblLookups. You can create multiple "Lookup-Types" which are essentially categories of lookup values the user can select from. For example, if one dashboard will



prompt a user to choose a facility, you would enter "Facility" in the "Lookup-Type" field in tblLookups for each value stored in the "Lookup-Value" field.

LookupID	Lookup-Type	Lookup-Value
110	Date-Range	All Time
111	Calculations	Sum
112	Calculations	Count
113	Calculations	Average
116	Date-Range	Exact Dates:
117	Field-Format	Currency (no decimals)
118	Field-Format	Number (no decimals)
119	Field-Format	Currency (2 decimals)
122	Field-Format	Number (2 decimals)
123	Field-Format	Number (all decimals)
124	Facility	West Division
125	Facility	East Division
126	Facility	Corporate
*	(New)	

Figure 2: Lookup Table Example

In the example shown in Figure 2 above, there are three facilities that a user can choose from. You might also have a dashboard that prompts the user to choose a Sales Manager's name, in which case you would group all the sales manager names by a different lookup type, like "Manager."

3.1.4.2 Associating Lookup Filters with Dashboards

The next step is to associate your chosen lookup filter with a dashboard. As shown in Figure 3 below, you select a "Lookup-Type" value in the "Required Lookup?" field within the Dashboard Administration form for a specific dashboard.

Enable Refresh Timer? ☒

Required Lookup? Facility VB Launch Code for this dashboard: floadDashboard 9

Figure 3: Dashboard Administration Screen

In the case above, it contains "Facility," which means any time a user opens this dashboard, they need to choose one of the available facilities.

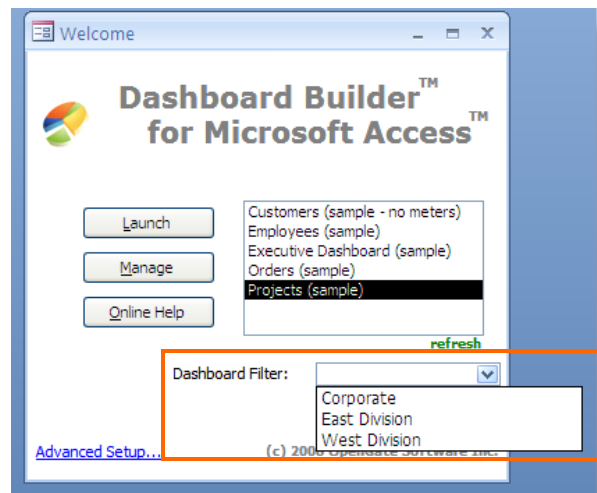


Figure 4: Dashboard Filter Example

Important!

Please read section 3.2.2.2 of this document regarding how to set up a metric to use the lookup filter a user selects.

3.2 CREATING METRICS

To open the Metric Administration form, select “Edit Metrics” from the bottom of the Dashboard Administration screen. Or, to edit a specific metric shown on screen, double click the name in the Dashboard Administration screen.



Figure 5: Dashboard Administration Screen



3.2.1 BASIC METRIC SETUP

Creating or modifying a metric is done in the Metric Definition Setup screen, as shown in Figure 3 below.

Figure 6: Metric Definition Screen

3.2.1.1 Metric Types

Dashboard Builder supports three metric types, described below:

- Average** Dashboard Builder will average all values contained in the Value Field specified for the Data Source and Date Range you provide.
- Count** Dashboard Builder will count the number of records that match the Date Range you provide in the Data Source.
- Sum** Dashboard Builder will sum all values contained in the Value Field specified for the Data Source and Date Range you provide.



Maximum Dashboard Builder will obtain the maximum value contains in the Value Field specified for the Data Source and Date Range you provide.

Minimum Dashboard Builder will obtain the minimum value contains in the Value Field specified for the Data Source and Date Range you provide.

3.2.1.2 Data Sources

Select the source of the data you want to use for the metric. Dashboard Builder supports Access tables and queries that reside in your database, or are linked from another Access database, Microsoft® SQL Server™ linked tables, and MySQL® server linked tables.

3.2.1.3 Date Ranges

Select from one of the predefined date ranges, use the "All Time" setting to collect all data regardless of date, or "Exact Dates:" to input From and To Dates. If you do not specify a "To" date value, Dashboard Builder will collect all data from the "From Date" value forward.

3.2.1.4 Value Field

The value field is the database field that contains the values you want to average or sum. For example, if you are averaging sales price, you may need to input "Total_Sales_Price" in the value field. For Metric Type "count" you simply need to provide the name of any field in the table.

3.2.1.5 Date Field

The date field is the database field that contains the date values you want to use when collecting your data. For example, you may choose to specify "Order-Date" for one metric, and "Ship Date" for another.

3.2.1.6 Validation

Before a metric can be chosen in the Dashboard Administration form, you must validate the setup. Valid metrics will appear with a green check in the upper right hand corner of the form, while invalid metrics will display a yellow warning box. When you validate a metric, it will display the current value that would be returned if the metric is valid. If the metric is invalid, Dashboard Builder will attempt to point out where the setup needs to be corrected.



3.2.2 ADVANCED SETTINGS

3.2.2.1 Display Formats

Choose from one of several different ways to display your data, including currency and numbers with or without decimals.

3.2.2.2 SQL-Where Statement

Provide a valid SQL string beginning with "WHERE..." Leave the field blank to select all records that meet the basic criteria you define. To employ the Dashboard Filter feature discussed in section 3.1.4 of this document, you will need to include the keyword "[GLOBALFILTER]" in your SQL statement where you want the user's chosen filter value to replace the keyword. For example, the SQL-Where statement:

WHERE [Status] = 'Complete' AND [Facility] = '[GLOBALFILTER]'

Will attempt to replace [GLOBALFILTER] with the string value the user chose when launching the dashboard. The available string values were defined at the time the dashboard is set up as shown below:

Enable Refresh Timer? ☒

Required Lookup? Facility VB Launch Code for this dashboard: floadDashboard 9

Figure 7: Dashboard Administration Screen

3.2.2.3 Meter Dashboard Settings

If you plan to display a metric in the meter format, you must supply both the Red and Green Limit values. The Red Limit value will represent the red end of the meter (typically an undesirable value), and the Green Value will represent the green end the meter. Depending on the metric, the Green metric may be greater or less than the Red metric. For example:

Metric	Red Limit	Green Limit
Orders This Month	5	100
Product Returns YTD	20	0
Revenue This Month	\$100	\$5,000
Expense This Month	\$5,000	\$100

Note that only metrics that have a defined Red and Green Limit will be available to select in the meter section of the Dashboard Administration form.



3.2.2.4 Drilldown Forms

You can optionally allow users to drill down into further detail for a given metric by specifying a form that Dashboard Builder should open when the user clicks on a metric. For example, you may have a form, "Orders" that you want a user to be able to open to display only those records that match the criteria used to return the metric. Perhaps you have a metric "Orders – Year to Date" which, when clicked, will show the user only orders for the current year.

If no drilldown form is specified, the metric will not be clickable by the user.

Note

The drilldown feature is only able to function if you incorporate Dashboard Builder into your own existing database application.

3.3 LAUNCHING DASHBOARDS FROM MACROS OR CODE

To launch a dashboard from VB code or a macro, you will need to call the function **fLoadDashboard()**. You can copy the necessary VB code supplied in the field "VB Launch Code for this dashboard:" at the bottom of the Dashboard Administration form. This field will contain the appropriate dashboard identifier that is used to launch a particular dashboard.

4 Advanced Setup

4.1 LAUNCH A DASHBOARD ON STARTUP

You may select a dashboard to automatically load when Dashboard Builder starts up as shown in Figure 9 below. Note that only dashboards that do not require a user to select a value from the Dashboard Filter field at startup will be available in the dropdown list.

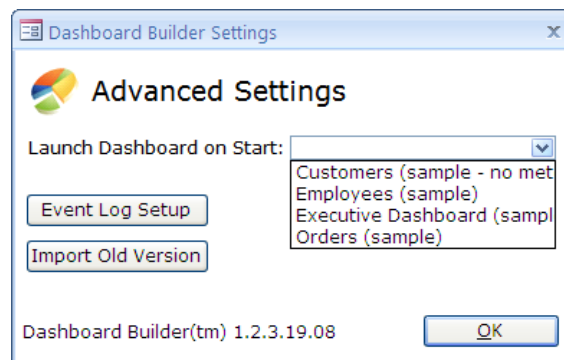


Figure 8: Advanced Setup Screen



4.2 CONFIGURING THE LOG

To configure the log, click on the Event Log Setup button, or open the form "frmLoggingAdmin" from the database window.

The screenshot shows a form titled "Administer Logging". It contains the following fields and controls:

- Logging-On:** A checkbox that is checked.
- Log Method:** A dropdown menu with "Table" selected.
- Log Location:** A text box containing "tblEventLog".
- Logging Level:** A dropdown menu with "Debug" selected.
- Event Emails:** A dropdown menu with "Never" selected.
- Email Admin:** A text box containing "test@test.com".
- OK button:** Located at the bottom right of the form.

Figure 9: Logging Administration Form

You can choose to turn logging on or off for the current Access database, specify if log events should be written to the event log table, or a file location you specify. Note that if you specify a file name, you will need to provide a fully qualified path name and file name in the "Log Location" field. The file format will be XML (without header/footer), and will not depend upon the file name extension you supply in the "Log Location" field.

The logging level will dictate whether certain events are written to the log or not. When you define a log event you can indicate if it is a Normal or Debug event. If you have the Logging Level set to "Debug," all events will be written to the log. If set to "Normal," only normal events will be written to the log, Debug events will be ignored.

Finally, you can send an email to a designated administrator when certain events occur. There are three values for the "Event Emails" setting:

Never – Events will never be emailed to the Administrator.

Defined Events – Only events where the parameter "blnEmailAdmin" is set to True in the fLogEvent() function.

Critical Errors – Any event where the fLogEvent() function parameter "intEventType" is set to "auCriticalError" will be emailed to the designated administrator.



5 Advanced Customization

While Dashboard Builder is primarily intended to function out of the box for users, experienced VB developers may be interested to leverage Dashboard Builder capabilities in their own customizations. This section describes how you can use Dashboard Builder functions to include dashboard components in your own forms.

5.1 ADDING METERS TO YOUR OWN FORMS

5.1.1 CREATE THE METER GROUP

To add one or more new meters to your own form, we recommend you copy four controls from either the form 'frmDashboard2a', or 'frmExamples'. The four controls, as shown in Figure 10 below, consist of the meter graphic, vertical black line, label 'Metric Name,' and unbound text box directly below the label.

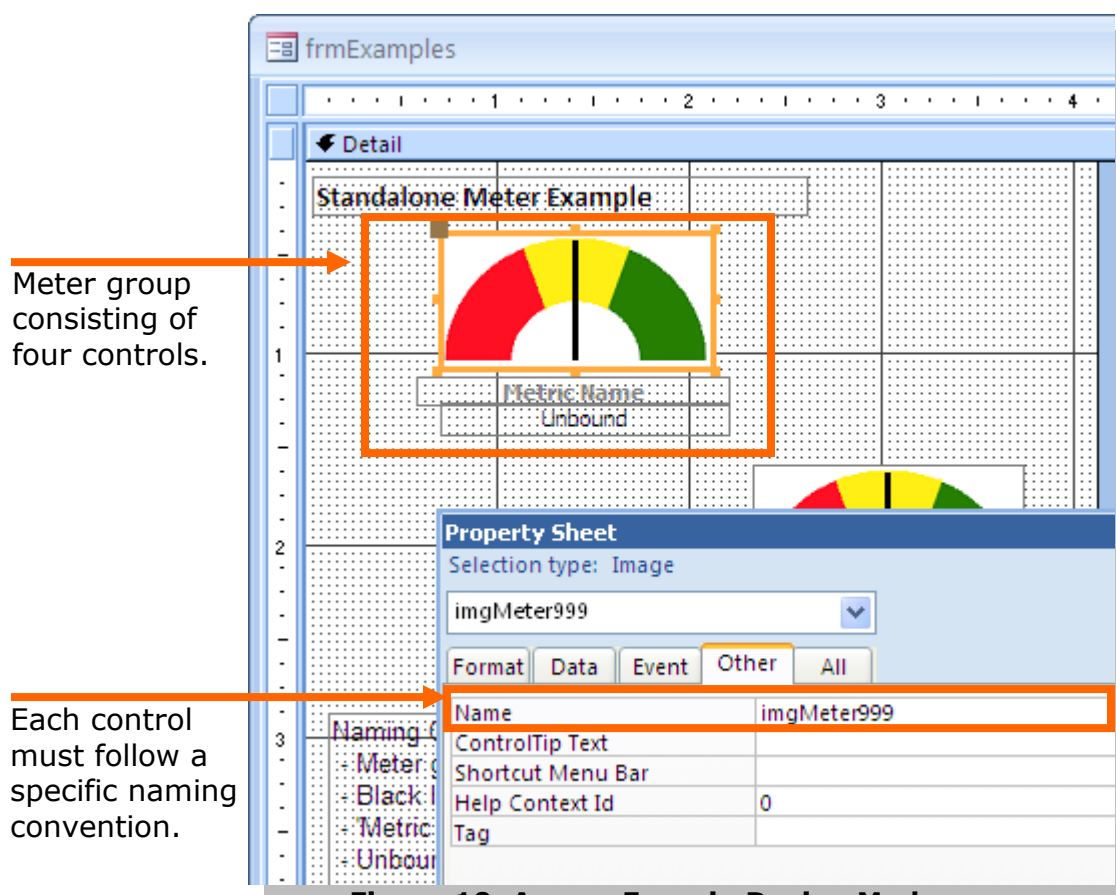


Figure 10: Access Form in Design Mode

Copy the four controls together as a group to your own form. If you are creating multiple meters on the same form, ensure they follow this naming convention:



Control	Naming Convention
Meter graphic	imgMeter###
Black line	Meter###
'Metric Name' label	lblGOM###
Unbound text field below 'Metric Name'	'valGOM###

Where ### is any numeric value from 0 to 32,767, and the same number should be used for each control in the group.

5.1.2 MODIFY THE FORM_LOAD EVENT

To set the meter to the appropriate value, and populate the labels, you must add a line of code to the Form_Load event on your form for each meter:

fSingleMeter iMetricID, iMeter, strFormName, blnRefresh

iMetricID – The metric ID from the table tblMetric that should be loaded for the current meter.

iMeter – The numeric value of the meter that is assigned to each control in the meter group.

strFormName – The form where the meter is located. "Me.Name" will be sufficient in most cases (except if you aren't calling fSingleMeter from the form itself).

blnRefresh – Optional. Set to False (default) if you do not want the meter to "spin-up" from 0 the value. Setting this value to False will speed up the meter display, and your form load.

In the following example, metric ID 53 will be loaded for meter 3 on the current form, and will display immediately (as opposed to "spinning-up" from zero):

fSingleMeter 53, 3, Me.Name

5.1.3 ENABLE DRILLDOWN AND MOUSE-OVER EFFECTS

To enable the ability for a user to drill-down into a form associated with a metric, you will need to two form events for each meter, and one subroutine.

- 1) Copy and paste the entire subroutine 'sMouseCursor' to your form, which can be found in the frmExamples form.
- 2) Add a '_MouseMove' event to each unbound text field 'valGOM###' with the following line:

sMouseCursor 0, ###



Where ### is the same number as the one you assigned to the meter group.

- 3) Add a '_Click' event to each unbound text field 'valG0M###' with the following line:

fDrillDown 0, 0, ###Me.Name, 49

Where ### is the same number as the one you assigned to the meter group, and replace 49 with the metric ID associated with the meter.

5.2 OBTAINING A METRIC VALUE FROM CODE

In some cases you may want to obtain the formatted value of a metric from code, without displaying that value on a dashboard. To do so, call the function fGetMetricValue in the following manner:

fGetMetricValue iMetricID, blnJustSQLStatement, lngDashboardID

iMetricID – The metric ID from the table tblMetric that should be returned.

blnJustSQLStatement – Optional. If this value is set to True, the function will return the constructed SQL where statement used by this metric, and not the value.

lngDashboardID – Optional. This value is needed only if the metric's SQL-Where statement includes the [GLOBALFILTER] keyword. The Dashboard ID identifies to fGetMetricValue which lookup value to prompt the user to provide if it is needed.

In the following example, metric ID 97 will be returned to the variable varReturnValue:

varReturnValue = fGetMetricValue(97)

6 Upgrading Dashboard Builder

You can upgrade from a previous version of Dashboard Builder in two ways, both of which are described in this section.



6.1 OPTION 1: IMPORT YOUR DASHBOARD AND METRIC DEFINITION TABLES INTO THE NEWEST VERSION OF DASHBOARD BUILDER.

In this scenario, you can navigate to the Advanced Setup screen from the Dashboard Builder Welcome screen, then select the “Import Old Version” button. You will be prompted to select your old version that contains your dashboard and metric configurations. Dashboard Builder will then import the following tables into the newest version of the application:

- tblDashboard
- tblDashboardMetric
- tblLookups
- tblMetric

6.2 OPTION 2: IMPORT ALL OF THE DASHBOARD BUILDER OBJECTS INTO YOUR EXISTING DATABASE.

If you have incorporated Dashboard Builder into an existing database, or linked numerous tables from other databases into your current version, you may wish to import the newest versions of the Dashboard Builder forms and VB modules. Be sure to import the following into your existing database, replacing any existing objects with the same name:

Forms

- frmDashboard1
- frmDashboard2a
- frmDashboardAdmin
- frmDBSetup
- frmDeleteDashboard
- frmHelp
- frmLoggingAdmin
- frmMetricAdmin
- frmPopup
- frmProcessing
- frmProgress
- frmWelcomeDashboard

VB Modules

- All VB code modules in the newest version



Release History

Dashboard Builder 1.0

- Initial release

Dashboard Builder 1.1

- Enhancements to the metric validation functionality.

Dashboard Builder 1.2

- New Advanced Setup screen.
- Ability to launch a dashboard automatically at the time Dashboard Builder is opened.

Dashboard Builder 1.3

- New dashboard filter feature allows you to prompt users to select a value prior to launching a dashboard. Any metrics in the dashboard that contain a keyword [GLOBALFILTER] in the SQL-Where statement will then use the user's selected value.
- Added upgrade capability.

Dashboard Builder 1.4

- Introduced new function fSingleMeter that allows you to place one or more meters on your own forms.
- Meters can now be positioned anywhere on a form (provided the appropriate naming conventions and controls are present).

Dashboard Builder 1.5

- Introduced new Maximum and Minimum metric types
- New ability to save, email, or print dashboards from the "print" icon at the bottom of each dashboard.