



Avaya Contact Center Select – Custom Reports Application Note

Overview

This application note applies to ACCS 6.4. It provides details on the installation of software required and tasks related to creating custom Historical Reports. The details covered in this application note are intended for advanced users.

ACCS Historical Reporting provides the ability to run reports based on the data stored on the ACCS server. The application is accessed via a web browser allowing the user to:

- Run reports immediately
- Schedule reports to be printed or exported
- Manage user defined versions of standard reports
- Import custom reports, which are available to the user via CCMA

The ACCS server is installed with a set of standard reports that cover the most important reporting scenarios for the Contact Center.

The solution doesn't support the Report Creation Wizard at this time, but it is possible to create new reports using external tools. This document describes a procedure that can be followed to build a custom report that is based on ACCS data.

IMPORTANT: The software required for creating ACCS custom reports must not be installed on the ACCS server. The installation process in this document is expected to be done on a client desktop.

1. Document Structure and Goals

This document is intended as a supplement to the existing ACCS documentation for advanced users. It specifically focuses on the software requirements and tasks involved in the creation of custom reports that will work with ACCS.

This document is not meant to provide an in-depth understanding of the tasks involved in the creation of custom reports; however, a limited set of details are provided. To create custom reports, the individual must have an understanding of:

- Writing SQL Select statements
- The ACCS database schema
- The ACCS Historical Reporting application
- The Microsoft SQL Server 2012 Report Builder 3.0 application

This document is structured as a guide to:

- Installing the required software
- Configuring ACCS database connections
- Performing the tasks to create a custom report

This document provides a step by step example of creating a custom report. Working through the example will help illustrate the specific requirements for integrating a custom report with ACCS.

1.1. Document References

1.1.1. AACC Documentation

This document references existing AACC documentation which is written to cover the features supported with the AACC solution. AACC documentation will contain details that are not applicable to ACCS. For example, the following items are not applicable to ACCS:

- CS1000 and AML
- CS2x00
- Networking
- Crystal Reports

1.1.2. Microsoft SQL Server Reporting Services

The help and online documentation related to SQL Server Reporting Services (SSRS) and Microsoft SQL Server 2012 Report Builder 3.0 include features not supported in the Report Viewer local mode processing, which is the implementation for ACCS. For example:

- Connections to non-ODBC data sources
- Ability to modify parameters while viewing reports
- Export formats other than PDF, Excel and Word
- SharePoint integration



- Data feeds and ATOM

2. ACCS Historical Reporting Overview

The ACCS solution uses the Microsoft SQL Server Reporting Services (SSRS) Report Viewer to generate, view, print and export reports. Historical Reporting uses the Report Viewer control in local mode processing. This requires the ACCS Historical Reporting module to retrieve the data and provide this to the Report Viewer control.

2.1. Report Format

The report format supported on ACCS is the Microsoft Report Definition Language (RDL) 2010. To build report files, the preferred software package available from Microsoft is:

- SQL Server 2012 Report Builder 3.0

The Report Builder generates report files with the extension RDL.

2.2. Custom Reports and Data Filtering

Custom reports can use the following features to filter the data that appears on a report when it is run through the CCMA interface:

- Data Range
- Data Partitioning

The Selection Criteria feature of Historical Reporting is not available for custom reports. If the report needs to be filtered beyond the Data Range and the filtering available from Data Partitioning, the filtering must be done within the report definition.

2.2.1. Data Range

The data range feature within CCMA allows the user to select the start and end date and time of the data included in the report. The data range feature is enabled if the report is imported as a Historical report. The data range feature is not available if the report is imported as a Configuration report.

Custom reports determine the Data Ranges that will be available to users during import to CCMA. The choices are:

- Interval
- Daily
- Weekly
- Monthly

2.2.2. Data Partitioning

ACCS custom reports can use the CCMA Data Partitioning feature. Data Partitioning will be used if the name of the Dataset on the report definition matches an existing CCMS or CCMM view name.



New reports must follow a set of rules in order to run successfully within CCMA. For example:

- The report can only use ODBC data sources
- For ACCS data sources, the data source name must follow the defined naming convention.
- The report must offer the SQL Select statement to retrieve the database data

3. Installing Required Software

The software required for creating ACCS custom reports must not be installed on the ACCS server. The following software is required:

- Microsoft SQL Server 2012 Report Builder
- Intersystem Cache ODBC driver

The report designer software must be downloaded. The Cache ODBC driver is available on the ACCS DVD.

3.1. Installing Microsoft SQL Server 2012 Report Builder

3.1.1. Before you begin

Ensure you have a web browser that can access external web pages. The software must be downloaded from Microsoft.

The software license restricts the use of the Microsoft SQL Server 2012 Report Builder to users that are licensed to use any one of the following editions of Microsoft SQL Server 2012 software:

- SQL Server 2012 Enterprise
- SQL Server 2012 Business Intelligence
- SQL Server 2012 Standard
- SQL Server 2012 Developer

These details are provided here to help explain the requirements before downloading the software; however this is not the official license agreement details. When installing the software, read the details provided before accepting the agreement.

Microsoft SQL Server items listed are not required for ACCS.

3.1.2. About this task

The report format supported on ACCS is the Microsoft Report Definition Language (RDL) 2010. This is the only software package available from Microsoft for building these files.

3.1.3. Procedure

1. Click or navigate to the following web page:
<http://www.microsoft.com/en-ie/download/details.aspx?id=29072>
2. Expand the **System Requirements** section and confirm that your system meets the requirements
3. Click the **Download** button
4. When requested, it is recommended to save the download file to a specific location.
5. Navigate to the downloaded file and double click the file to start the installation.
6. On the **Welcome** screen, click **Next**

7. On the **License Agreement** screen, read the agreement. Select the option **I accept the terms in the license agreement** and click the **Next** button or click the **Cancel** button to stop the installation.
8. On the **Feature Selection** screen, change the install location if required, and click the **Next** button.
9. On the **Default Target Server** screen enter the SQL Server details following the examples provided and click the **Next** button. If not connecting to the SQL Server, leave the details blank and click the **Next** button
10. On the Ready to Install the Program screen, click the Install button.
11. When the installation completes, click the **Finish** button.

3.2. Installing the Caché ODBC driver

3.2.1. Before you begin

Ensure you have the most recent version of the Contact Center DVD.

3.2.2. About this task

To run custom reports or SQL queries from client computers you must first install an Intersystems Caché ODBC driver on the client computer. You can either install 32-bit or 64-bit driver based on the system configuration on your client's computer. The choice between 32-bit and 64-bit depends on:

- Operating System being 32-bit or 64-bit
- Application using the ODBC driver being 32-bit or 64-bit

The Intersystems Caché ODBC drivers are available on the Avaya Aura Contact Center Installation DVD.

The Report Builder 3.0 application is a 32 bit application, so the 32-bit ODBC driver must be installed. If planning to access the databases via a 64-bit application (e.g. Microsoft Excel 64 bit), you must also install the 64-bit driver.

Note:

Install this driver only on client computers. Do not install this driver on Contact Center servers.

3.2.3. Procedure

1. Insert the Contact Center Release DVD into your client computer.
2. Navigate to the `ThirdParty\Cache ODBC Driver\` folder. This folder contains 32-bit and 64-bit drivers.
3. Double click the appropriate Intersystems Caché ODBC driver application:

Version	File Name	Installation
32-bit	ODBCDriver_2007.1_x86.exe	Mandatory
64-bit	ODBCDriver_x64.exe	Optional



This starts the InstallShield Wizard for Intersystems ODBC driver.

4. Click Next on the InstallShield Wizard for Intersystems ODBC driver.
5. Click Install to install the driver.
6. When the driver is installed, click Finish.

4. Configuring Database Connections

ACCS reporting supports ODBC connections only. If the ODBC DSN name starts with a known server type followed by an underscore character, the connection will be updated dynamically at runtime to match the connection details for the specific server type in ACCS configuration. This allows the same report to be imported to multiple ACCS systems.

The following chart shows the supported types and naming convention of the ODBC Connections:

Server Type	ODBC	Recommended
CCMS	Starts with CCMS_	CCMS_PREVIEW_DSN
CCMM	Starts with CCMM_	CCMM_PREVIEW_DSN

The recommended value shown is the same ODBC DSN used on the standard reports shipped with ACCS. The tasks in this document will assume the recommended value has been used.

The report must include a Dataset using a connection to one of the supported database servers or the report will not run. This allows for connections to other ODBC databases to appear on the report, and these do not need to be configured in ACCS. ODBC connections to other databases are not dynamically managed, meaning they will always connect using the provided ODBC connection. As well, this same ODBC connection must be manually configured on each ACCS server where the report is imported.

4.1. Defining the Caché ODBC DSN for CCMS

4.1.1. Before you begin

Make sure the Intersystems Caché port 1972 is open in the client computer firewall. Select a desktop user account that the DSN uses to log into the Contact Center Manager Server database. If needed, ask an administrator for the account details. For more information on adding desktop user accounts, see Avaya Aura® Contact Center Server Administration (44400-610).

4.1.2. About this task

To access the Contact Center Manager Server database, you must define a connection to the server and define the Caché ODBC DSN.

4.1.3. Procedure

- From the **Windows Start** menu, choose **Run**. Enter one of the following in the **Open** text box
 - OS and ODBC driver are both 32-bit or 64-bit:
C:\Windows\System32\odbcad32.exe
 - OR
 - OS 64-bit and ODBC driver 32-bit: C:\Windows\SysWOW64\odbcad32.exe
- Click OK to launch the ODBC Data Source Administrator dialog.
- Click the **System DSN** tab.
- Click **Add**.

5. Select Intersystems ODBC.
6. Click Finish. The system displays the Intersystems Cache ODBC Data Source Setup dialog.
7. Enter the following information
 - a. **Name:** Enter the required name of the DSN. The recommended value is `CCMS_PREVIEW_DSN`
 - b. **Description:** (Optional) Enter a description of the DSN
 - c. **Host:** Enter the IP address or computer name of the Contact Center Manager Server
 - d. **Port:** Confirm that the port number is 1972
 - e. **Cache Namespace:** Enter `CCMS_STAT`
 - f. **User Name:** Enter the user name of the desktop user account
 - g. **Password:** Enter the password of the desktop user account
8. Click **Test Connection** to test the connection.
9. Click **OK**.

4.2. Defining the DSN for CCMM

4.2.1. Before you begin

Make sure the Intersystems Caché port 1972 is open in the client computer firewall. The CCMM Administration application can be used to change the Reporting Credentials. If needed, ask an administrator for the account details.

By default, the reporting credentials are:

User Name: mmReport

Password: mmRep

4.2.2. About this task

To access the Contact Center Multimedia database from a report writer application, you must install the Caché ODBC driver and define the DSN.

4.2.3. Procedure

1. From the **Windows Start** menu, choose **Run**. Enter one of the following in the **Open** text box:
 - a. OS and ODBC driver both 32-bit or 64-bit:
`C:\Windows\System32\odbcad32.exe`

OR

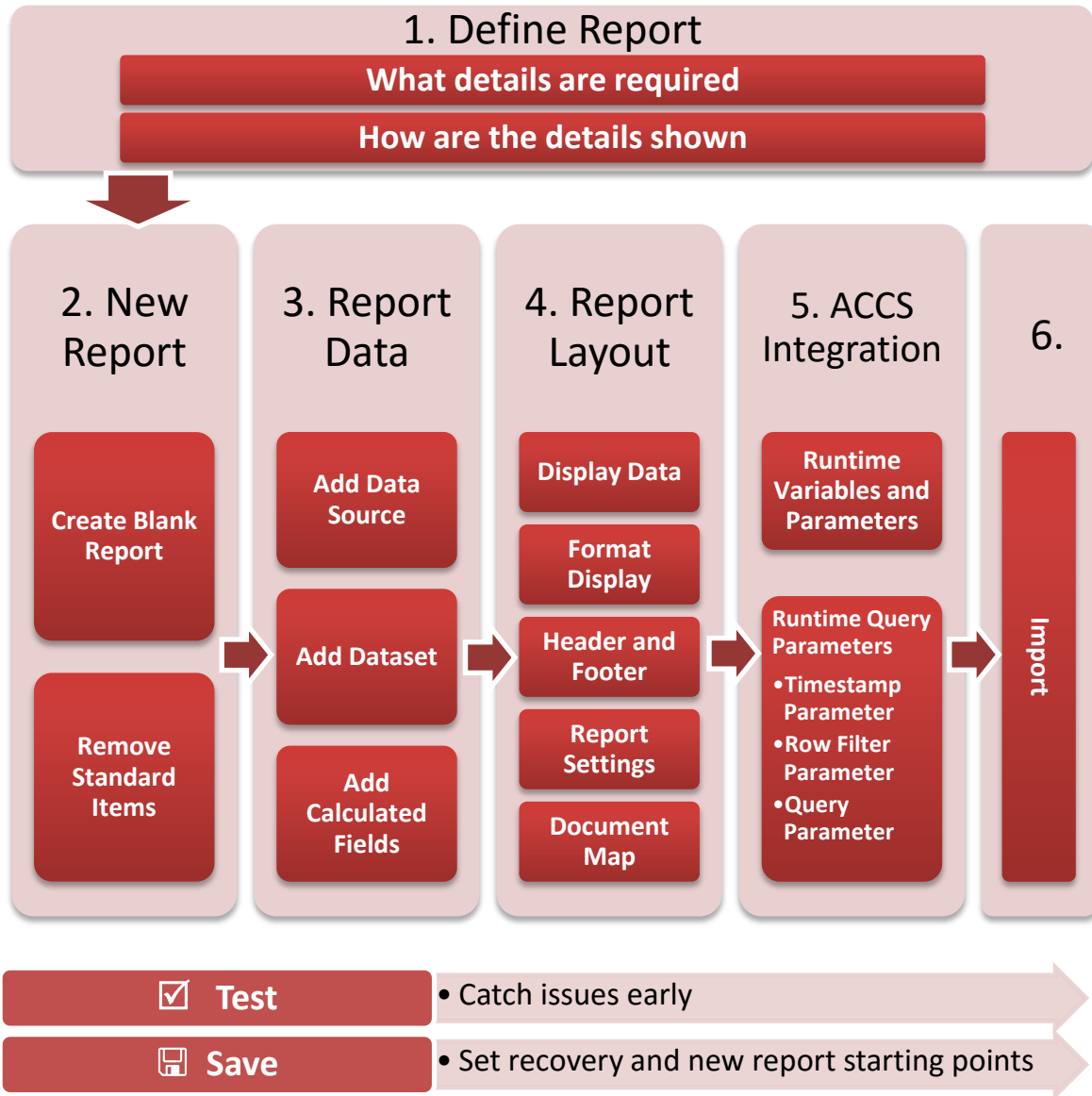
 - b. OS 64-bit and ODBC driver 32-bit: `C:\Windows\SysWOW64\odbcad32.exe`
2. Click OK to launch the ODBC Data Source Administrator dialog.
3. Click the **System DSN** tab.
4. Click **Add**.



5. Select Intersystems ODBC.
6. Click Finish. The system displays the Intersystems Cache ODBC Data Source Setup dialog.
7. Enter the following information
 - a. **Name:** Enter the required name of the DSN. The recommended value is
CCMM_PREVIEW_DSN
 - b. **Description:** (Optional) Enter a description of the DSN
 - c. **Host:** Enter the IP address or computer name of the Contact Center Multimedia Server
 - d. **Port:** Confirm that the port number is 1972
 - e. **Cache Namespace:** Enter MULTIMEDIA
 - f. **User Name:** Enter the reporting credentials Account ID (i.e. mmReport)
 - g. **Password:** Enter the reporting credentials password (i.e. mmRep)
8. Click **Test Connection** to test the connection.
9. Click **OK**.

5. Creating a custom report

This section describes the steps for creating a custom report for use with ACCS. The following shows an overview of the tasks involved:



Testing and saving the report should be done often while creating a report. Testing ensures problems are seen around the same time a change is made, making it easier to determine the issue or undo the change. Saving provides a recovery point, and if saved under different names, starting points to produce a different report with similar properties.

5.1. Define Report

The best way to start building a report is to have an idea of the details required and how these details are shown.

5.1.1. Required Details

The report must be based on database data and can include items such as:

- Report Title
- Report user details
- Page count

5.1.1.1. Related Documentation

The AACC Performance Management Data Dictionary provides details on the CCMS and CCMM database schemas. See the following chapters for details:

- CCMS: Avaya Aura Contact Center Manager Data dictionary
- CCMM: Contact Center Multimedia Data dictionary

The guide also provides details on the standard reports included with Historical Reporting. See the following chapter for details:

- Standard Reports

For each of the standard reports, the following details are included:

- Database views used for the report
- Database fields or formulas used on the report

Since the guide is written for the AACC solution, more standard reports will be available in this guide than available with ACCS Historical Reporting. The file names shown use the extension RPT, the ACCS report file will have the same name with the extension is RDL.

5.1.2. Presentation Options

The report data can be shown as text, graphically or both. The data can be shown as detailed rows, it can be grouped, and reports can also show summarized values for groups and report totals.

5.2. New Report

5.2.1. Blank Report

The process outlined starts with a blank report, the Data Source and Dataset are added as separate steps. This allows the report to be saved as a starting point for building additional reports that share the same data.

5.2.2. Remove Standard Items

The standard items that exist by default on a new report will not be used and should be removed.

5.3. Report Data

5.3.1. Add Data Source

A Data Source defines the connection details used to retrieve data. Datasets are associated to a Data Source so at least one Data Source must be defined before adding a Dataset.

5.3.2. Add Dataset

A Dataset is associated to a Data Source and defines the data fields available.

To use ACCS Data Partitioning, the Dataset name must match the CCMS or CCMM view name. For more information, see [Data Partitioning](#) Data Partitioning in [Appendix A: ACCS Databases](#).

A report query, which is an SQL Select statement, is required. For details, see [Appendix B: Writing the Report Query](#).

5.3.3. Add Calculated Fields

A dataset calculated field is calculated for each row of data returned from the report query. Although it is possible to use the SQL Select statement to produce similar results, the expression engine in SSRS provides a standard set of functions which are not database server specific.

Use a calculated field when:

- Groups are based on an expression and not a database field
- The same expression is used multiple times on the report

5.4. Report Layout

5.4.1. Display Data

The SSRS reports offer several ways of showing the data. For summarized data, such as a dashboard, the report can simply include graphics and text based elements.

To show the individual records and group summaries, the report must include a tablix. A tablix is a common report item capable of displaying data in a Table, Matrix or List. A tablix can show text and graphical items.

5.4.1.1. Table

A table displays detail data or grouped data, or a combination of both. Data is formatted in rows and columns, and groups can display group totals.

A table typically grows horizontally and not vertically, so it is possible to ensure a table fits a page width; however, the number of pages will increase as the table increases horizontally on the report.

5.4.1.2. Matrix

A matrix, like a pivot table, allows groups to be assigned to rows and columns. A matrix displays aggregate values (Sum, Max, Min, Average, etc.) in cells scoped to the intersection of the row and column.

A matrix grows horizontally and vertically on the page. A matrix can span multiple pages if the width exceeds the page width or the height exceeds the page height. The report viewer will display the width as a single page. When the report is printed or exported, the page width will be enforced and the matrix will span multiple pages.

5.4.1.3. List

A list repeats with each group or row in the report data. A list can be used to create free-form reports or forms. Lists can contain any number of report items and can be nested within another list.

A list typically grows horizontally and not vertically, so it is possible to ensure a list fits a page width; however, the number of pages will increase as the list increases horizontally on the report.

5.4.1.4. Available Graphics

The available graphics include:

- Column, Bar Pie, Line and Area Charts
- Gauges
- Maps
- Data Bar
- Column, Line, Area, Shape, Range Sparkline
- Directional, Symbols, Shapes and Ratings Indicators

The Data Bar, Sparkline and Indicators graphics represent small versions of the Charts and Gauges. They are meant as graphics that can appear within a Table, Matrix or List beside cells showing text values. They allow a report to combine text representation and graphical representation of the data in the same row or column.

5.4.1.5. Groups and Group Totals

To display groups of data on the report, text and graphics must be placed in a Tablix. The Tablix defines data regions. These regions provide subsets of data which are used to produce the aggregated results.

Each group can have a row or multiple rows placed above and below the next group or details of the data records. These can be thought of as group headers and footers, and these sections repeat each



time the group changes. Typically groups would show the details of the group and aggregate data for the specific group value.

5.4.2. Format Display

Items added to the report will be formatted using default settings. The report items displayed can be formatted to change how values are display, adjust colors, add borders, etc.

In the case of ACCS data fields, Timestamp fields and elapsed time fields require specific format settings.

5.4.3. Header and Footer

The report header is a section of the report repeated at the top of each page. Determine if the report will have a report header and what items will be included in the header.

The report footer is repeated at the bottom of each page. Determine if the report will have a report footer and what will be included in the footer.

5.4.4. Report Settings

The report settings determine the printed and exported page size and orientation, print margins, background color, etc. Use the report settings to refine the report.

5.4.5. Document Map

A document map provides a set of navigational links to report items. A user can click links in the document map to jump to the report page that displays that item.

5.5. ACCS Integration

These are important integration points between the custom reports and the ACCS Historical Reporting runtime infrastructure. If the report is not going to use this functionality, this section can be skipped.

For details, see [Appendix E: ACCS Integration](#).

5.5.1. Runtime Variables and Parameters

Runtime variables and parameters have their values replaced at runtime with values defined in Historical Reporting. Variables are not available when writing expressions for calculated fields for a Dataset. Parameters are available when writing expressions for calculated fields for a Dataset.

5.5.2. Runtime Query Parameters

Runtime query parameters are required if:

- Report will be filtered based on the user selected Data Range
- Report will be filtered based on the user's Data Partitioning

A runtime query is built from a set of parameters that have their values read at runtime. These parameters do not have their values replaced at runtime.

5.6. Import

When a report is imported, the report is saved on the Contact Center server. The report is accessed by opening the Private Report Templates folder. To share the report with other users, a copy of the report must be saved in a Group Folder.

For details, see **Importing a report** in **Administering Avaya Contact Center Select** from the standard documentation.

5.7. Testing Reports

The Report Builder offers the ability to view the report as it will appear in the Report Viewer, on the printed page and to export the report. The Report Builder offers more export formats than available with the Report Viewer. The 3 formats supported by the Report viewer, and therefore ACCS, are PDF, Excel and Word.

5.7.1. Test using Report Viewer

1. With the report loaded in the Report Builder, click the **Home** tab
2. Click the **Run** icon
3. If required, enter the **User Name** and **Password** for the report connection
4. When the report loads, navigate the pages and ensure the report is appearing correctly. Look for word wrapping, truncated items, value formats etc.

5.7.2. Test Print Layout

1. With the report loaded in the Report Builder, click the **Home** tab
2. Click the **Run** icon
3. If required, enter the **User Name** and **Password** for the report connection
4. When the report loads, click the **Page Setup** icon and select either Portrait or Landscape for the **Orientation**. This setting is for the viewer and is not saved to the report. If the report requires Landscape orientation, ensure the Page Size is set correctly as Landscape.
5. Click **Print Layout** icon if not already selected
6. Navigate the pages and ensure the report is correctly fitting the printed page.
7. Click the **Print Layout** icon again to return to the Report Viewer

5.7.3. Test Exported

To test the exported reports the associated applications for each export type must be available.

1. With the report loaded in the Report Builder, click the **Home** tab
2. Click the **Run** icon



3. If required, enter the **User Name** and **Password** for the report connection
4. When the report loads, click the **Export** icon and select an export format
5. When the exporting completes, save the file to a known location.
6. Navigate to the exported file and double click the file to open it in the associated application.
7. Review the exported report and ensure the format is correct.

6. Example - Daily Agent Performance Report

6.1. Define Report

6.1.1. Required Details

The report must be based on database data. The details for this example are:

- ODBC Connection: CCMS_PREVIEW_DSN
- View: dAgentPerformanceStat
- Fields:

```

Timestamp
AgentLogin
AgentGivenName
AgentSurName
CallsOffered
CallsAnswered
LoggedInTime
TalkTime
HoldTime
PostCallProcessingTime
NotReadyTime
WaitingTime
    
```

6.1.2. Presentation Options

The example report will be built to show the following details in each section of the report:

Section	Description of items shown
Header	<ul style="list-style-type: none"> • Report Title • Data Range selected by user and used in report query
Body	<ul style="list-style-type: none"> • Table representation of data: <ul style="list-style-type: none"> ○ Data grouped by Agent Name and ID, then Date ○ Group and report totals
Footer	<ul style="list-style-type: none"> • File path • User that ran the report and when the report was generated • Page count

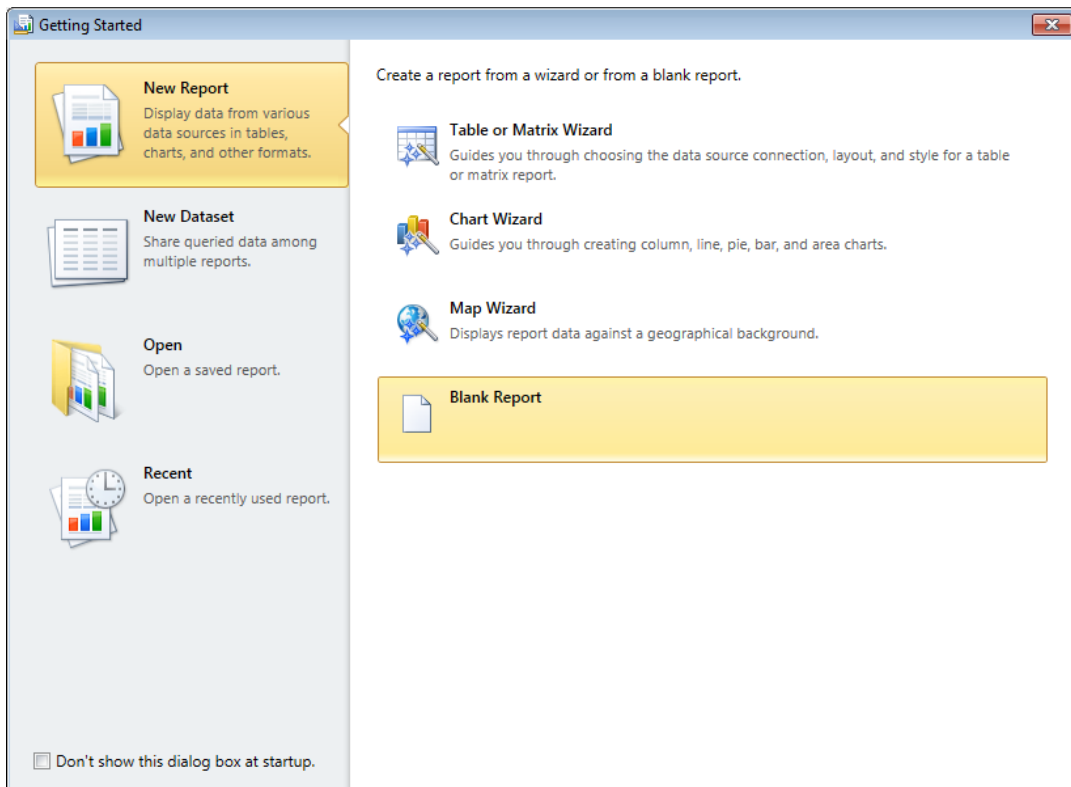
6.2. New Report

6.2.1. Create Blank Report

This example starts with a blank report, the Data Source and Dataset are added as separate steps. This allows the report to be saved as a starting point for building additional reports that share the same data.

6.2.1.1. Procedure

1. Launch Report Builder 3.0
2. On the **Getting Started** dialog select **Blank Report** under **New Report**



6.2.2. Remove Standard Items

The new report will automatically have a Text box for adding a report title, and another for the execution time in the report footer. These text boxes will not be used and this procedure shows how to remove them.

6.2.2.1. Before you begin

Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

6.2.2.2.Procedure

1. Click the **Text Box** with the text **Click to add title**. When correctly selected the **Properties** view will show **ReportTitle** as the selected item.
2. Press the **Delete** key to delete the text box.
3. Click the **Text Box** with the text **[&ExecutionTime]**. When correctly selected the **Properties** view will show **ExecutionTime** as the selected item.
4. Press the **Delete** key to delete the text box.

6.3. Report Data

The report must be based on some data available from a data source. The report defines the database details as a Data Source and a Dataset. The Dataset can also contain Calculated Fields as in this example.

6.3.1. Add Data Source

A Data Source defines the connection details used to retrieve data. Datasets are associated to a Data Source so at least one Data Source must be defined before adding a Dataset.

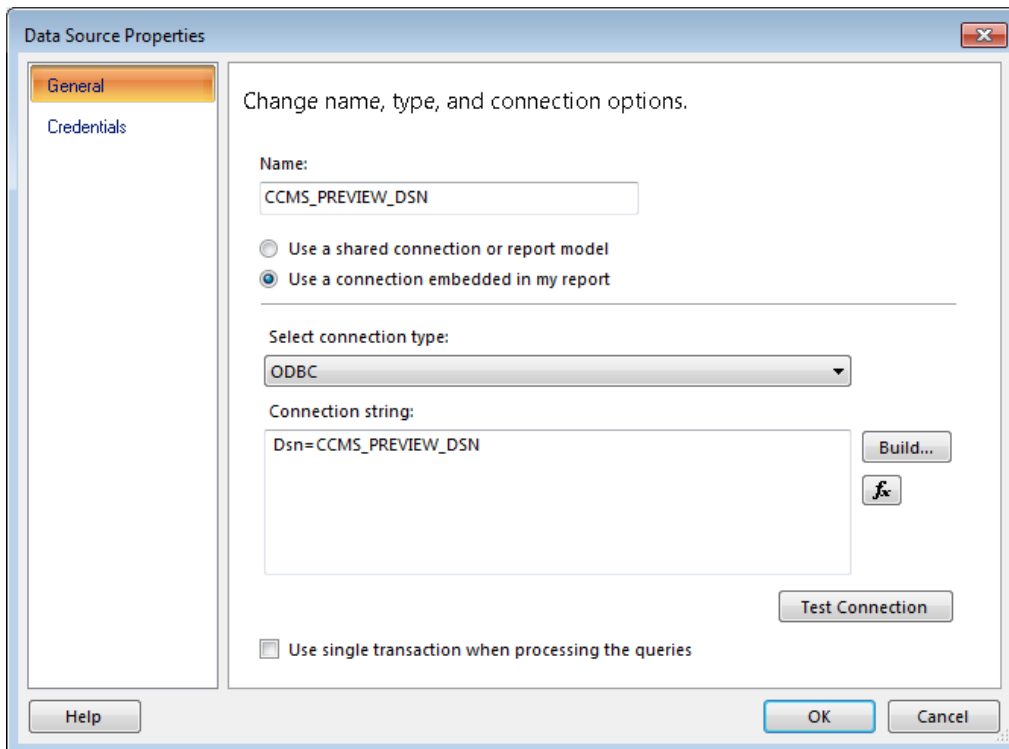
The Data Source will be the CCMS_PREVIEW_DSN ODBC Connection.

6.3.1.1.Before you begin

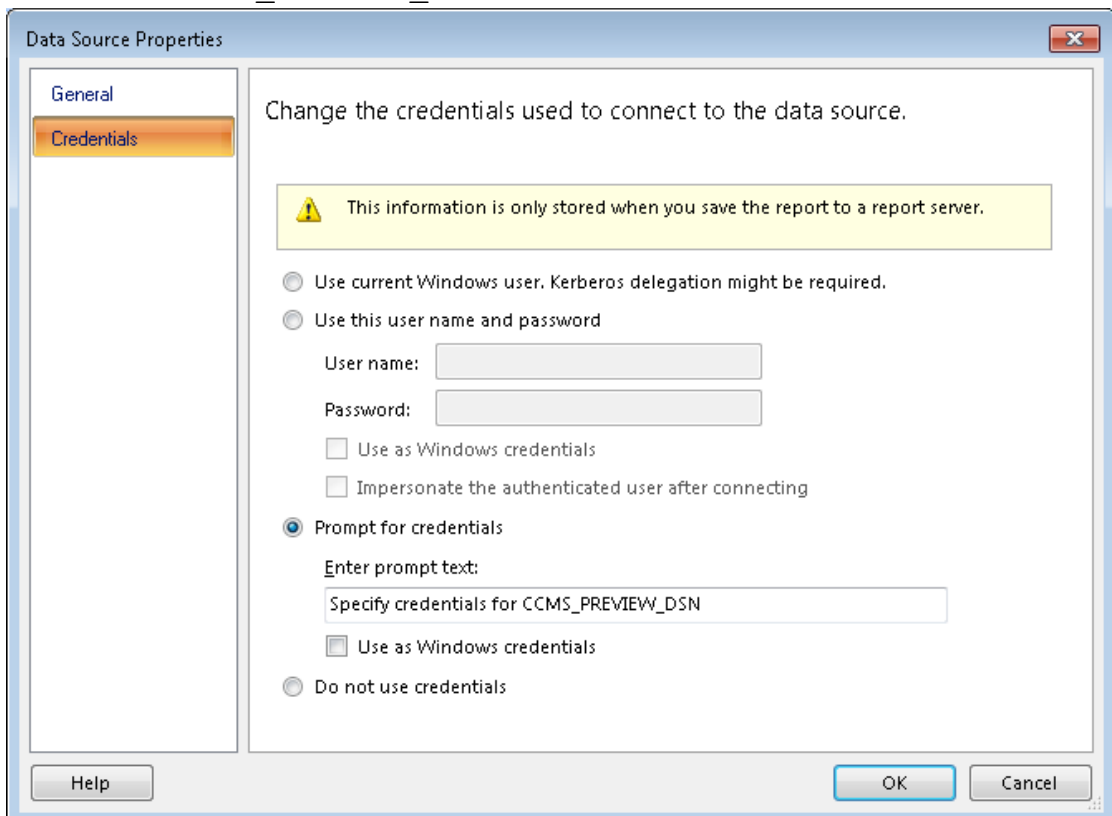
Make sure that the **Report Data** tree view is visible. If **Report Data** is not visible select the **View** menu item and check the box for **Report Data**.

6.3.1.2.Procedure

1. Right click **Data Sources** in the **Report Data** tree view and select **Add Data Source ...** from the pop up menu.
2. Select **General** in the **Data Source Properties** dialog
 - a. Enter CCMS_PREVIEW_DSN in the **Name** field.
 - b. Select the option **Use a connection embedded in my report**
 - c. Select **ODBC** from the dropdown list for **Select connection type**
 - d. Click **Build ...** beside Connection string text box.
 - e. Select **Use user or system data source name** and select CCMS_PREVIEW_DSN from the drop down list under **Data source specification** in the **Connection Properties** dialog.



- f. Click **OK** to accept the connection properties and close the **Connection Properties** dialog.
3. Select **Credentials** in the **Data Source Properties** dialog
 - a. Select the **Prompt for credentials** option and enter Specify credentials for CCMS_PREVIEW_DSN.





4. Select **General** in the **Data Source Properties** dialog.
 - a. Click **Test Connection**, enter the **User name** and **Password** for the connection. Prompting for credentials ensures they are not saved with the report.
5. Click **OK** to create the data source and close the **Data Source Properties** dialog.

6.3.2. Add Dataset

A Dataset is associated to a Data Source and defines the data fields available.

The Dataset name will be `dAgentPerformanceStat` which will allow Data Partitioning to be used. For more information, see [Data Partitioning](#) Data Partitioning in [Appendix A: ACCS Databases](#).

The Dataset for this example will be based on the report query from [Updated Agent Performance Statistics Example](#) in [Appendix B: Writing the Report Query](#).

6.3.2.1. Before you begin

Make sure that the **Report Data** tree view is visible. If **Report Data** is not visible select the **View** menu item and check the box for **Report Data**.

6.3.2.2. Procedure

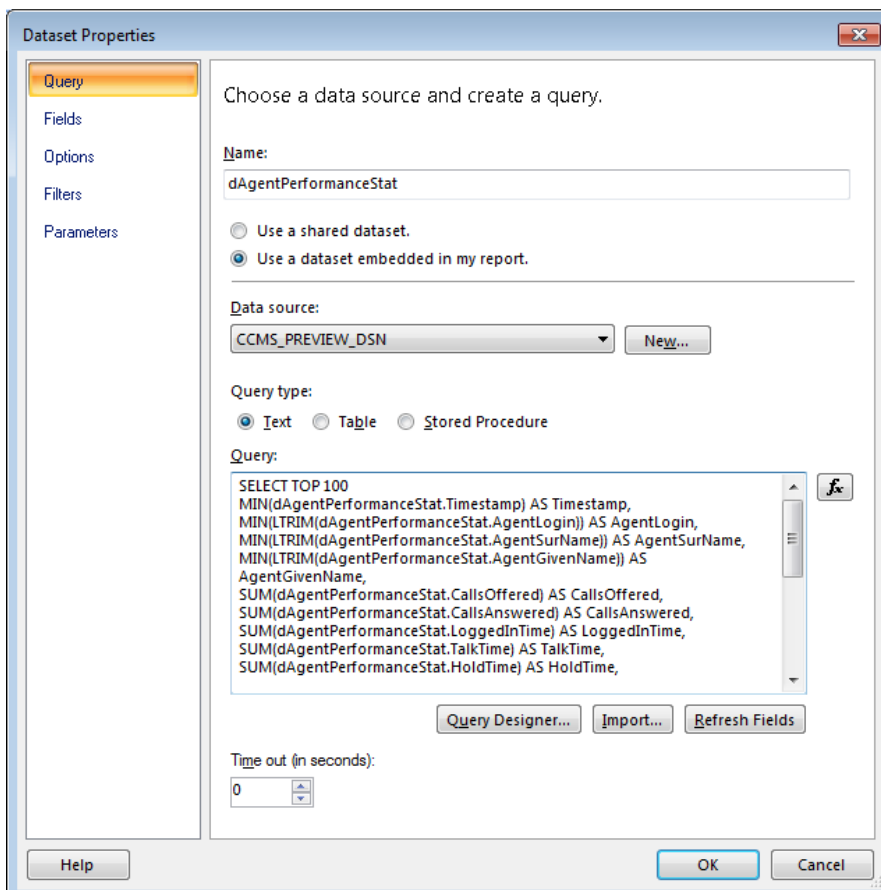
1. Right click **Datasets** in the **Report Data** tree view and select **Add Dataset ...** from the pop up menu.
2. Select **Query** in the **Dataset Properties** dialog
 - a. Enter `dAgentPerformanceStat` in the **Name** field.
 - b. Select the option **Use a dataset embedded in my report**
 - c. Select **CCMS_PREVIEW_DSN** from the dropdown list for **Data source**
 - d. Select **Text** for **Query type**
 - e. Paste the following SQL Select statement in the **Query** field:

```
SELECT TOP 100
MIN(dAgentPerformanceStat.Timestamp) AS Timestamp,
MIN(LTRIM(dAgentPerformanceStat.AgentLogin)) AS
AgentLogin,
MIN(LTRIM(dAgentPerformanceStat.AgentSurName)) AS
AgentSurName,
MIN(LTRIM(dAgentPerformanceStat.AgentGivenName)) AS
AgentGivenName,
SUM(dAgentPerformanceStat.CallsOffered) AS CallsOffered,
```

```

SUM(dAgentPerformanceStat.CallsAnswered) AS
CallsAnswered,
SUM(dAgentPerformanceStat.LoggedInTime) AS LoggedInTime,
SUM(dAgentPerformanceStat.TalkTime) AS TalkTime,
SUM(dAgentPerformanceStat.HoldTime) AS HoldTime,
SUM(dAgentPerformanceStat.PostCallProcessingTime) AS
PostCallProcessingTime,
SUM(dAgentPerformanceStat.NotReadyTime) AS NotReadyTime,
SUM(dAgentPerformanceStat.WaitingTime) AS WaitingTime
FROM dbo.dAgentPerformanceStat dAgentPerformanceStat
WHERE (dAgentPerformanceStat.Timestamp>= DateAdd('d', -8,
Now()) And dAgentPerformanceStat.Timestamp<= DateAdd('d',
-1, Now()))
GROUP BY dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin
ORDER BY dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin

```



3. Click **OK** to create the dataset and close the **Dataset Properties** dialog.
4. If required, input data source credentials in **Enter Data Source Credentials** popup window. Click **OK**.

6.3.3. Add Calculated Fields

A dataset calculated field is an expression calculated for each row of data returned from the report query. Although it is possible to use the SQL Select statement to produce similar results, the expression engine in SSRS provides a standard set of functions which are not database server specific.

The Dataset contains fields for AgentGivenName, AgentSurName and AgentLogin. To group and show the “Agent Name – ID”, a calculated field will be added.

For details on Expressions, see the online help included with the Report Builder.

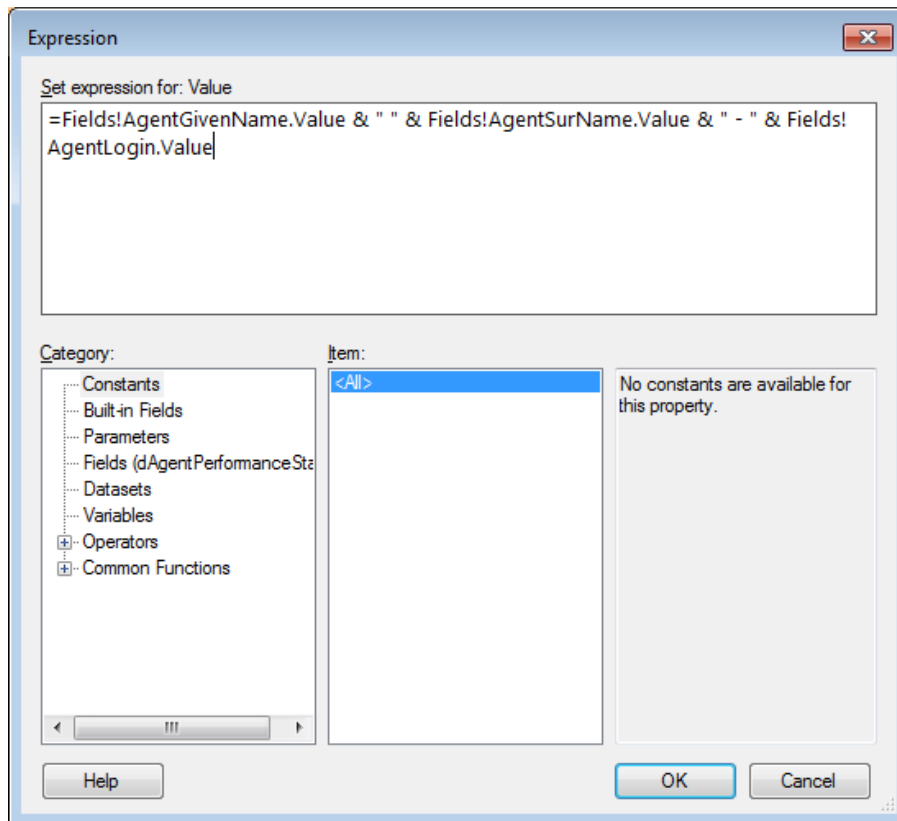
6.3.3.1. Before you begin

Make sure that the **Report Data** tree view is visible. If **Report Data** is not visible select the **View** menu item and check the box for **Report Data**.

6.3.3.2. Procedure

1. Right click dAgentPerformanceStat under **Datasets** in the **Report Data** tree view and select **Add Calculated Field ...** from the pop up menu.
2. A blank row is added to the end of the **Fields** list in the **Dataset Properties** dialog.
3. Click the empty text box under the Field Name column and enter Cal_AgentNameID
4. Click the **fx** button beside the empty text box under the **Field Source** column
 - a. Select the text in the **Set expression for: Value** field and press the **Delete** key to remove all text
 - b. Paste the following Agent Name - ID expression into the **Set expression for: Value** field:

```
=Fields!AgentGivenName.Value & " " &  
Fields!AgentSurName.Value & " - " &  
Fields!AgentLogin.Value
```



- c. Click **OK** to accept the changes and close the **Expression** dialog
5. Click **OK** to accept the changes and close the **Dataset Properties** dialog
6. Test using report viewer. Even when the report is empty, testing the report will confirm that the work so far is valid.
7. Save the report. Suggested name `AgentPerformance_Dataset.rdl`

6.4. Report Layout

6.4.1. Display Data

The report will use a Table to display the data. The data will be grouped by Agent Name – ID and Timestamp, with summarized values included for each group and grand totals.

6.4.1.1. Table and Matrix Wizard

The Table and Matrix wizard simplifies the process of creating either a Table or Matrix.

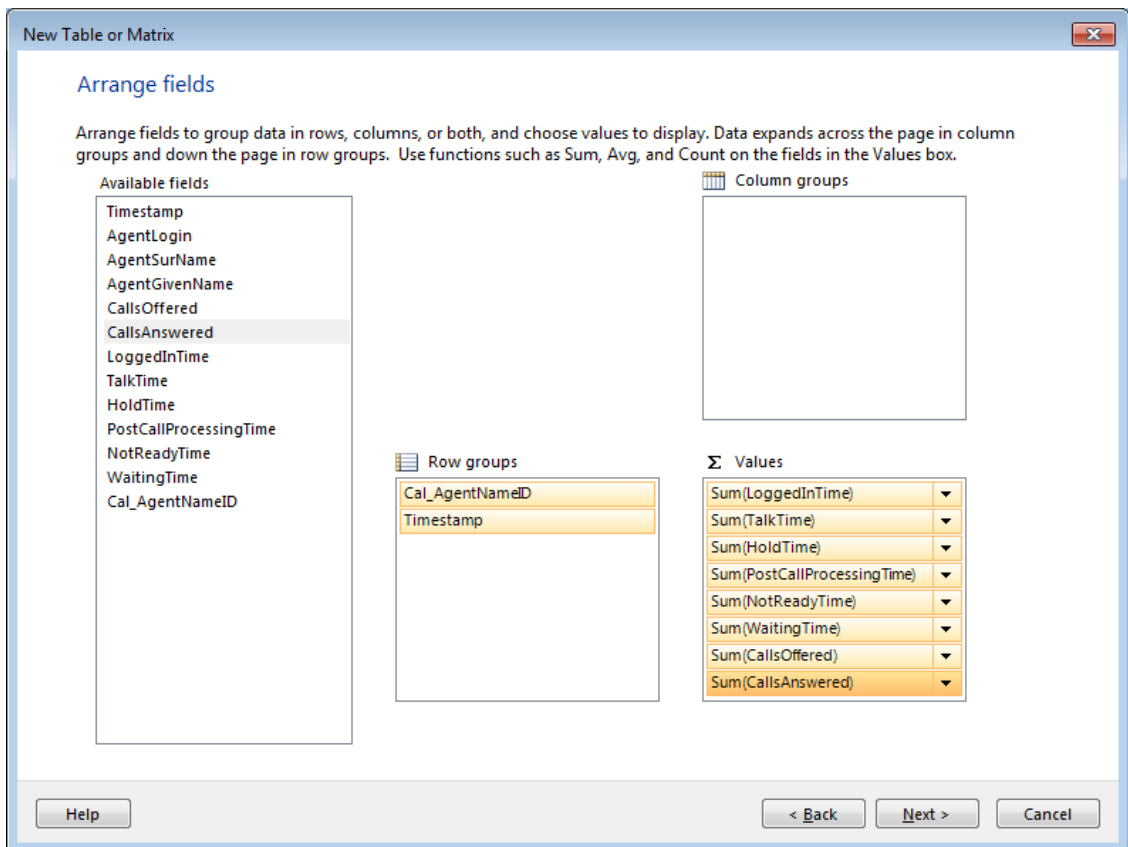
6.4.1.1.1. Before you begin

Make sure that the **Data Source and Dataset have been added to the report.**

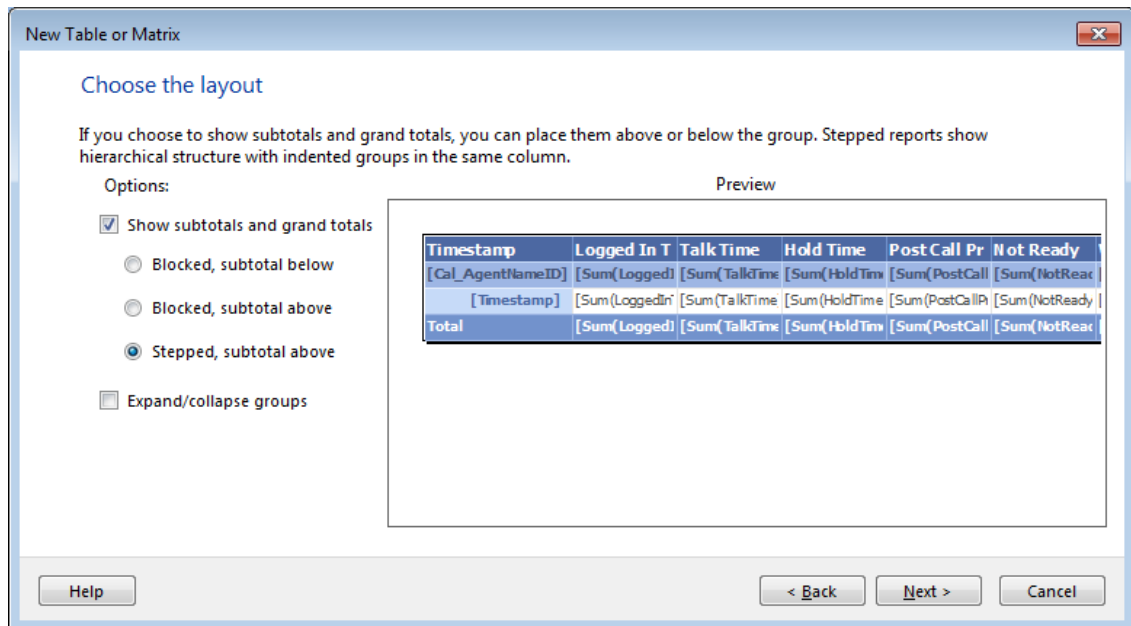
6.4.1.1.2. Procedure

1. Click the **Insert** menu.
2. Click the **Table** icon and select **Table Wizard**.
3. On the **Choose a dataset** page, select `dAgentPerformanceStat` from the list and click the **Next** button.

4. On the **Arrange fields** page:
 - a. Move items from the **Available fields** list as following:
 - i. **Row groups** list:
 - Cal_AgentNameID
 - Timestamp
 - ii. **Values** list:
 - LoggedInTime
 - TalkTime
 - HoldTime
 - PostCallProcessingTime
 - NotReadyTime
 - WaitingTime
 - CallsOffered
 - CallsAnswered



- b. Click the **Next** button.
5. On the **Choose the layout** page:
 - a. Check the **Show subtotals and grand totals** option, and select **Stepped, subtotal above**
 - b. Uncheck the **Expand/collapse groups** option.



- c. Click the **Next** button
6. On the **Choose a style** page, select **Ocean** from the **Styles** list.
7. Click the **Finish** button.
8. Test using report viewer.
9. Save the report. Suggested to save as AgentPerformance_Table.rdl

6.4.2. Format Display

Items added to the report will be formatted using default settings. In the case of ACCS data fields, Timestamp fields and elapsed time fields require specific format settings.

6.4.2.1. Format Timestamp Fields

The format required for a Timestamp field will depend on the details the report should offer. For this Daily report, only the Date part is required.

This procedure uses Microsoft standard format strings to ensure the report displays in the user preferred format. For details, see [Appendix H: Globalization](#)

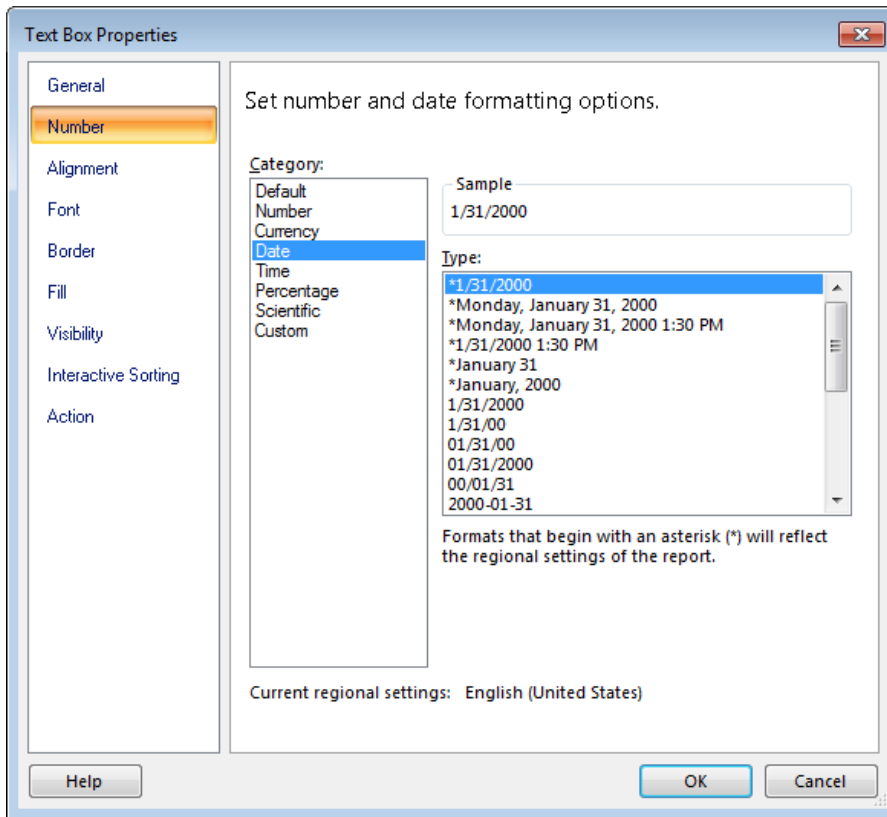
6.4.2.1.1. Before you begin

Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

6.4.2.1.2. Procedure

1. Click the cell with the text **[Timestamp]**. When correctly selected the **Properties** view will show **Timestamp** as the selected item.
2. Right click the cell and select **Text Box Properties ...** from the popup menu.
3. Select **Number** in the **Text Box Properties** dialog

- a. Select **Date** from the **Categories** list and select the short date format from the **Type** list. The short date format will be *1/31/2000 for US English. By selecting the format with the * the format will follow the users preferred format.

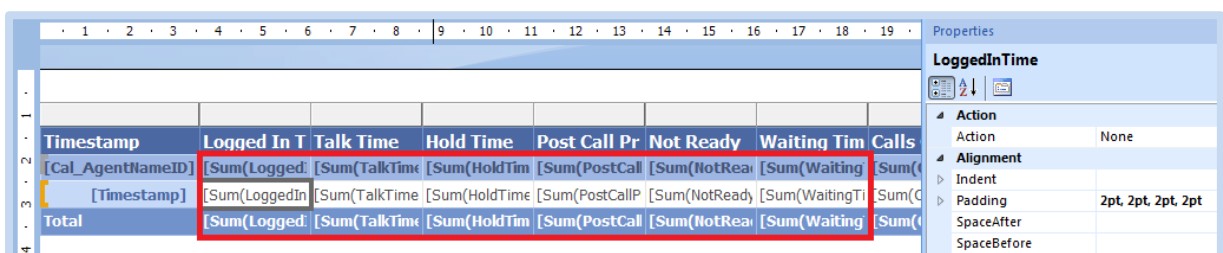


4. Click **OK** to accept the settings and close the **Text Box Properties** dialog.

6.4.2.2. Format Elapsed Time Fields

Elapsed time fields are numeric data fields storing the value in seconds. The Report Builder does not offer a format that can be directly applied to elapsed time numeric field (e.g. hh:mm:ss).

The following shows one of the elapsed time fields selected and the red box outlines the fields of the report that are elapsed time fields.



All of the expressions are the same for each of the summary cells in a column showing elapsed time. The following table shows the elapsed time expression that will replace each of the existing expressions:

Column Heading	Elapsed Time Expression
Logged In Time	=IIf(Sum(Fields!LoggedInTime.Value) = 0, "00:00:00", Format(Floor(Sum(Fields!LoggedInTime.Value) / 3600),"00") &":"& Format(TimeSerial(0,0, Sum(Fields!LoggedInTime.Value)), "mm:ss"))
Talk Time	=IIf(Sum(Fields!TalkTime.Value) = 0, "00:00:00", Format(Floor(Sum(Fields!TalkTime.Value) / 3600),"00") &":"& Format(TimeSerial(0,0, Sum(Fields!TalkTime.Value)), "mm:ss"))
Hold Time	=IIf(Sum(Fields!HoldTime.Value) = 0, "00:00:00", Format(Floor(Sum(Fields!HoldTime.Value) / 3600),"00") &":"& Format(TimeSerial(0,0, Sum(Fields!HoldTime.Value)), "mm:ss"))
Post Call Processing Time	=IIf(Sum(Fields!PostCallProcessingTime.Value) = 0, "00:00:00", Format(Floor(Sum(Fields!PostCallProcessingTime.Value) / 3600),"00") &":"& Format(TimeSerial(0,0, Sum(Fields!PostCallProcessingTime.Value)), "mm:ss"))
Not Ready Time	=IIf(Sum(Fields!NotReadyTime.Value) = 0, "00:00:00", Format(Floor(Sum(Fields!NotReadyTime.Value) / 3600),"00") &":"& Format(TimeSerial(0,0, Sum(Fields!NotReadyTime.Value)), "mm:ss"))
Waiting Time	=IIf(Sum(Fields!WaitingTime.Value) = 0, "00:00:00", Format(Floor(Sum(Fields!WaitingTime.Value) / 3600),"00") &":"& Format(TimeSerial(0,0, Sum(Fields!WaitingTime.Value)), "mm:ss"))

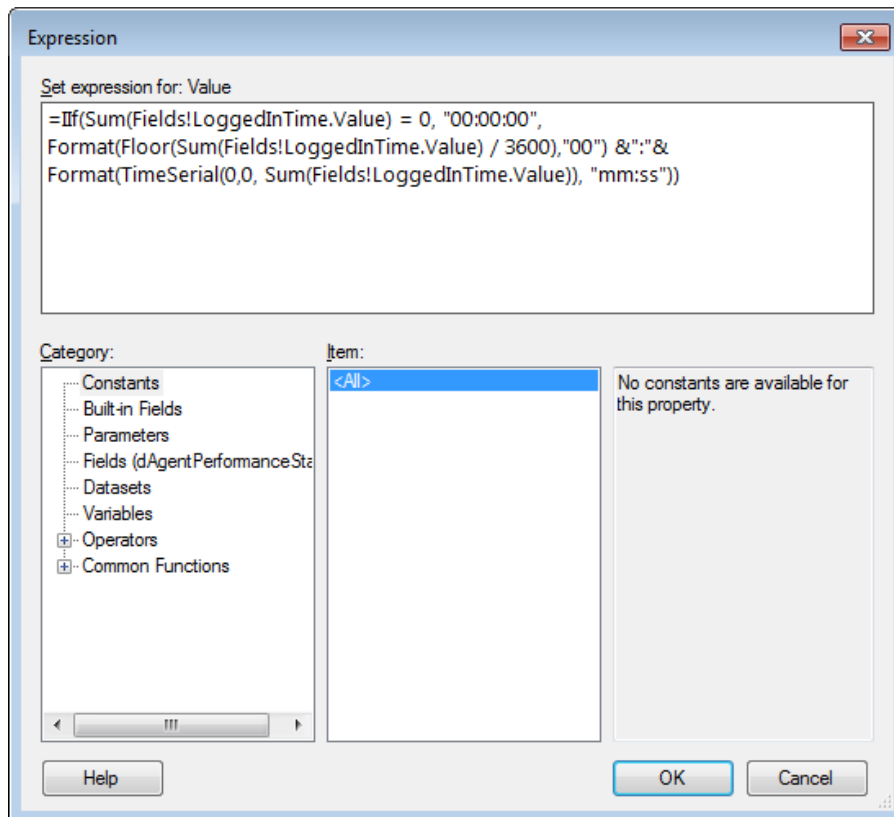
For details on Expressions, see the online help included with the Report Builder.

6.4.2.2.1. Before you begin

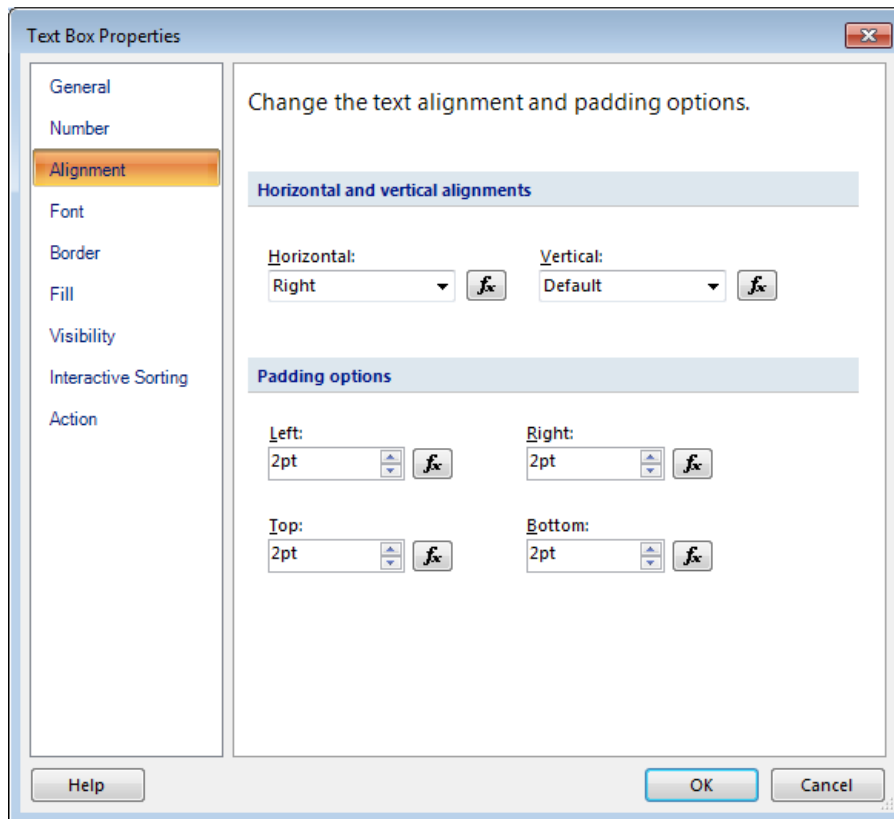
Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

6.4.2.2.2. Procedure

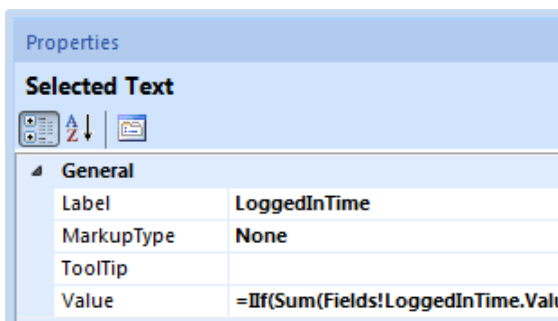
1. Click the first cell with the text **[Sum(LoggedInTime)]**. When correctly selected the **Properties** view will show **LoggedInTime** or **TextboxNN** (where NN is a number) as the selected item.
2. Right click the cell and select **Expression ...** from the popup menu.
 - a. Select the text in the **Set expression for: Value** field and press the **Delete** key to remove all text.
 - b. Paste the elapsed time expression text from the table above into the **Set expression for: Value** field. For example, for each of the cells under the Logged In Time column, the elapsed time expression is:




3. Click **OK** to accept the changes and close the **Expression** dialog
4. Right click the cell and select **Text Box Properties ...** from the popup menu.
 - a. Select **Alignment** in the **Text Box Properties** dialog
 - b. Select the **Right** from the drop down list under **Horizontal**.



5. Click **OK** to accept the changes and close the **Text Box Properties** dialog
6. Click the Text <<Expr>>. When correctly selected the **Properties** view will show **Selected Text**.
7. In the **Properties** view, select the **Categorized** icon and locate the **General** category.
8. Enter a name for the **Label** value, such as the field name (e.g. LoggedInTime)



9. Repeat steps 1 to 12 for each of the cells under the "Logged In Time" heading.
10. Test using report viewer.
11. Repeat steps 1 to 14 for each of the remaining cells with elapsed time values, which currently display:
 - [Sum(TalkTime)]
 - [Sum(HoldTime)]
 - [Sum(PostCallProcessingTime)]
 - [Sum(NotReadyTime)]

- [Sum(WaitingTime)]
12.  Save the report.

6.4.2.3. Repeat Table Headings on Each Page

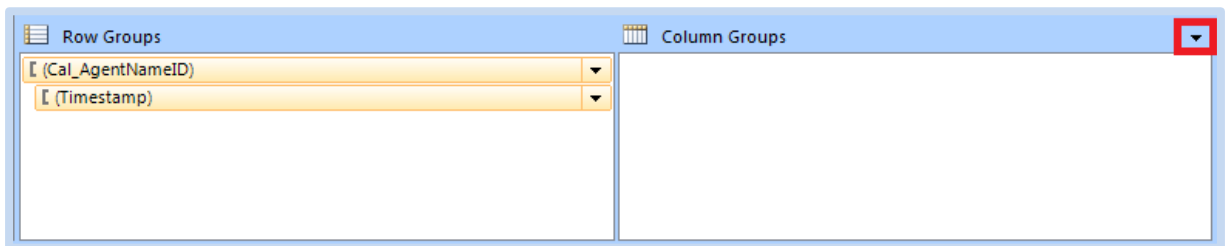
The Table and Matrix Wizard will create a row that shows the column heading names. The row is not repeated on each page by default.

6.4.2.3.1. Before you begin

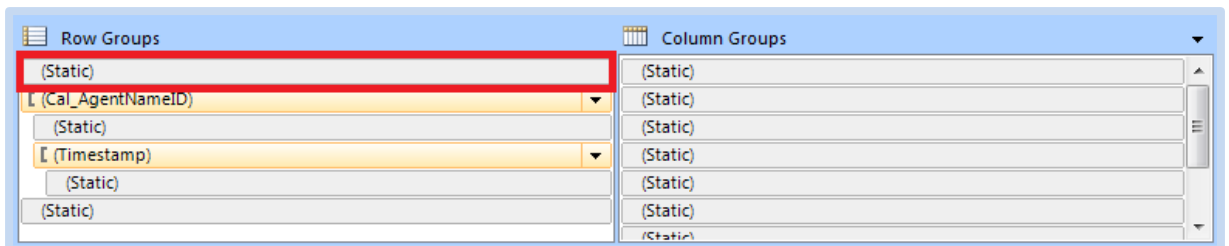
Make sure that the **Properties** and **Grouping** views are visible. If **Properties** or **Grouping** is not visible select the **View** menu item and check the boxes for **Properties** and **Grouping**.

6.4.2.3.2. Procedure

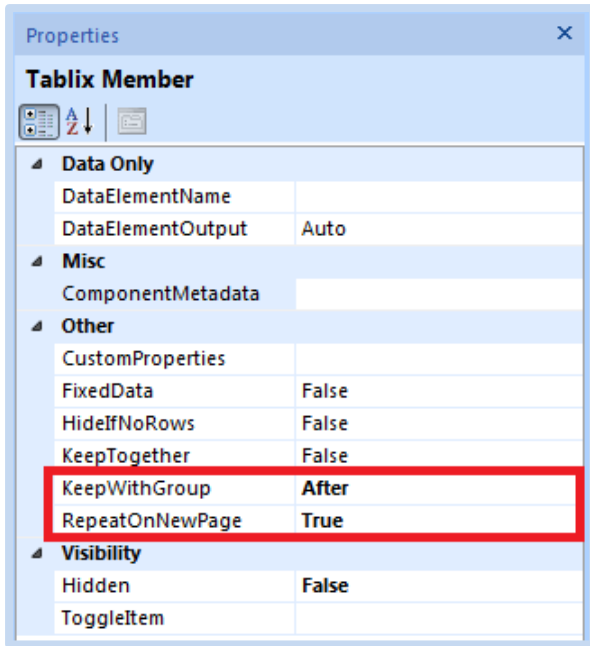
1. Click the arrow beside the **Column Groups** in the **Grouping** view and select **Advanced Mode**.



2. Click the first **Static** item under the **Row Groups**.



3. In the **Properties** view:
 - a. Select **After** from the drop down list beside **KeepWithGroup**
 - b. Select **True** from the drop down list beside **RepeatOnNewPage**



4. Test using report viewer.
5. Save the report.

6.4.3. Header and Footer

The report header is a section of the report repeated on every page. The report footer is repeated at the bottom of each page.

The header and footer of this example will show details passed from the ACCS Historical Reporting application. This is part of the ACCS integration; however for the example it makes sense to define these before adding items to the Header and Footer section.

The following runtime and global values will be used:

Collection	Item	Description
Variables	report_title	The report title defined in Historical Reporting.
Variables	report_filepath	The file path of the report
Variables	report_user	The name of the user running the report
Parameters	report_interval	An internationalized string representing the collection period for the report.
Globals	ExecutionTime	The date and time that the report began to run.
Globals	OverallPageNumber	The page number of the current page for the entire report.

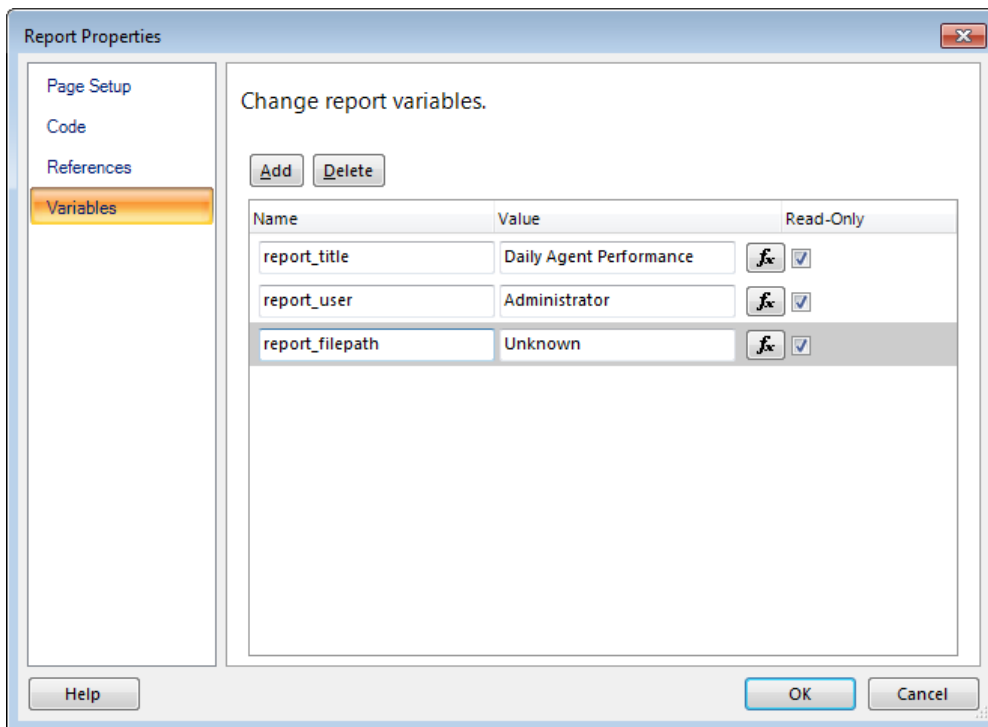
6.4.3.1. Add Runtime Report Variables

Runtime variables have their values replaced at runtime with values defined in Historical Reporting. The Variables collection is not available when writing expressions for calculated fields for a Dataset.

For details on available report variables, see [Runtime Variables](#) in [Appendix E: ACCS Integration](#).

6.4.3.1.1. Procedure

1. Right click the mouse off the visible report page (i.e. Above, Below, Left or Right) and select **Report Properties ...** from the popup menu
2. Select Variables in the **Report Properties** dialog
3. Click the **Add** button and enter:
 - a. **Name:** report_title
 - b. **Value:** Daily Agent Performance
4. Click the **Add** button and enter:
 - a. **Name:** report_user
 - b. **Value:** Administrator
5. Click the **Add** button and enter:
 - a. **Name:** report_filepath
 - b. **Value:** Filepath Placeholder



6. Click **OK** to accept the changes and close the **Text Box Properties** dialog
7. Test using report viewer.
8. Save the report. Suggested name AgentPerformance_HeaderFooter.rdl

6.4.3.2. Add Runtime Report Parameters

Runtime parameters have their values replaced at runtime with values defined in the ACCS Historical Reporting subsystem. The Parameters collection is available when writing expressions for calculated fields for a Dataset.

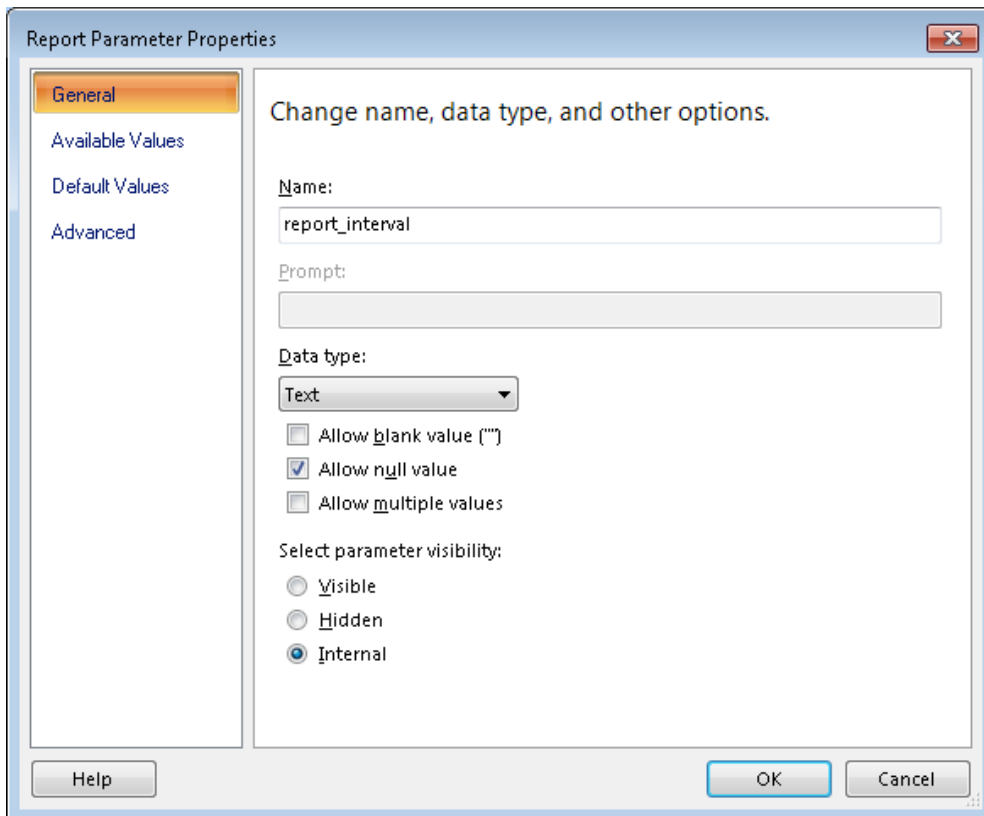
For details on the available report parameters, see [Runtime Parameters](#) in [Appendix E: ACCS Integration](#).

6.4.3.2.1. Before you begin

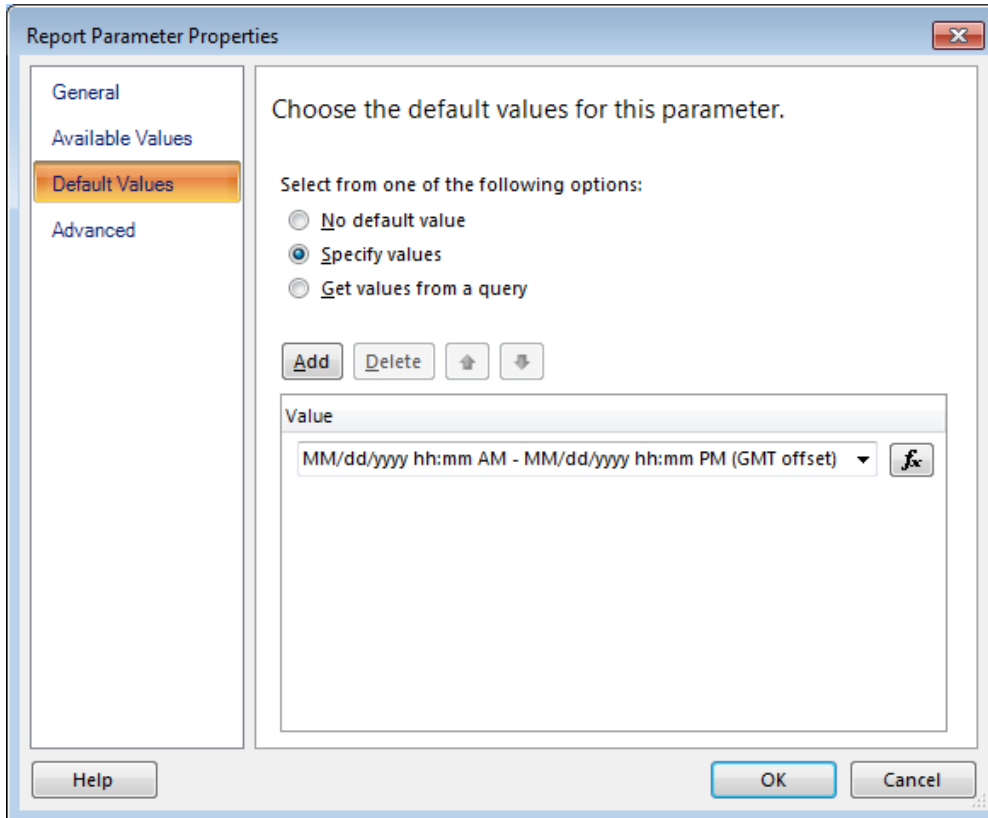
Make sure that the **Report Data** tree view is visible. If **Report Data** is not visible select the **View** menu item and check the box for **Report Data**.

6.4.3.2.2. Procedure

1. Right click **Parameters** in the **Report Data** tree view and select **Add Parameter ...** from the pop up menu
2. Select **General** in the **Report Parameter Properties** dialog
 - a. Enter `report_interval` in the **Name** field.
 - b. Select **Text** for **Data Type** from the drop down list
 - c. Check the **Allow null value** check box
 - d. Select **Internal** under **Select parameter visibility**



3. Select **Default Values** in the **Report Parameter Properties** dialog
 - a. Select **Specify values** under **Select from one of the following options**
 - b. Click the **ADD** button and enter the following:
 - i. Value: `MM/dd/yyyy hh:mm AM - MM/dd/yyyy hh:mm PM (GMT offset)`.
This value can be anything, and the value shown is the US English format of the value. Entering the expected format string will provide an understanding of the space required on the report.



4. Click **OK** to accept the changes and close the **Report Parameter Properties** dialog
5. Test using report viewer.
6. Save the report.

6.4.3.3. Add Report Header

New Reports do not contain a Report Header. The items included in this example are the runtime values:

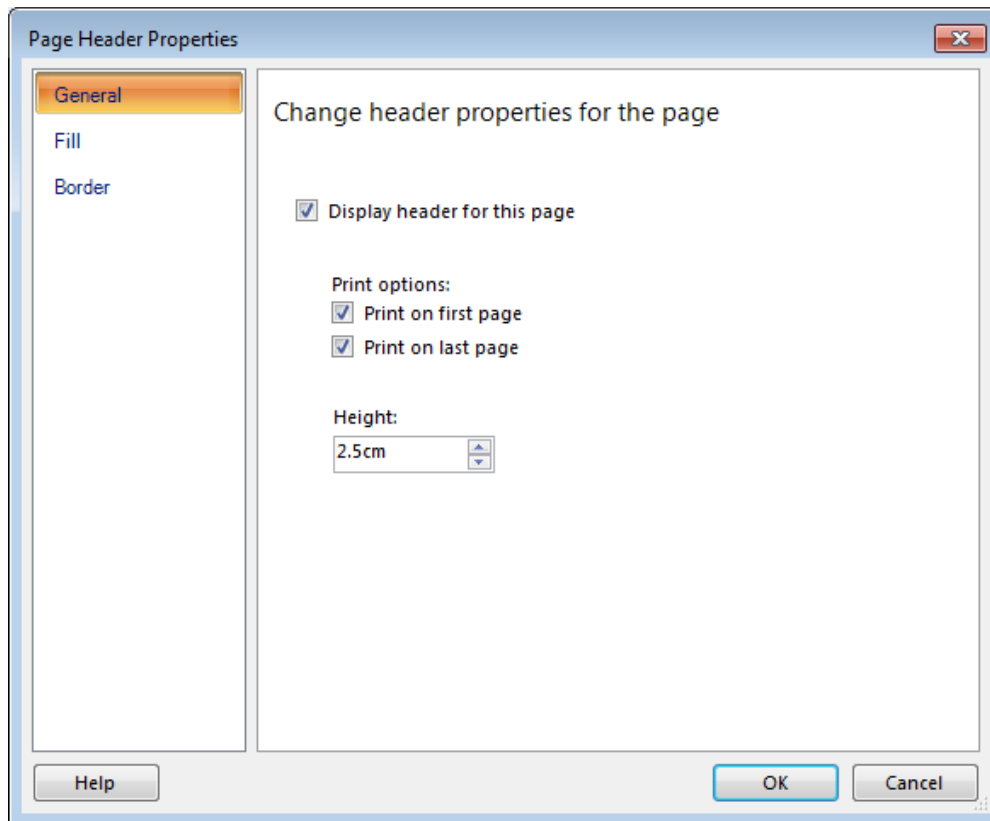
- report_title
- report_interval

6.4.3.3.1. Before you begin

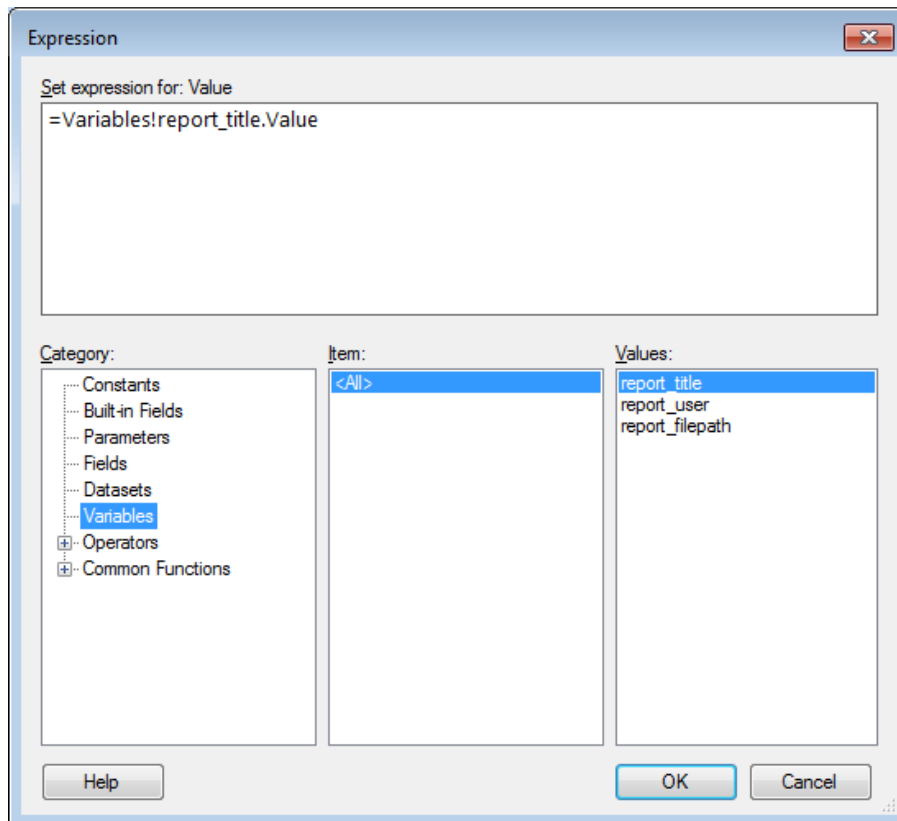
Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

6.4.3.3.2. Procedure

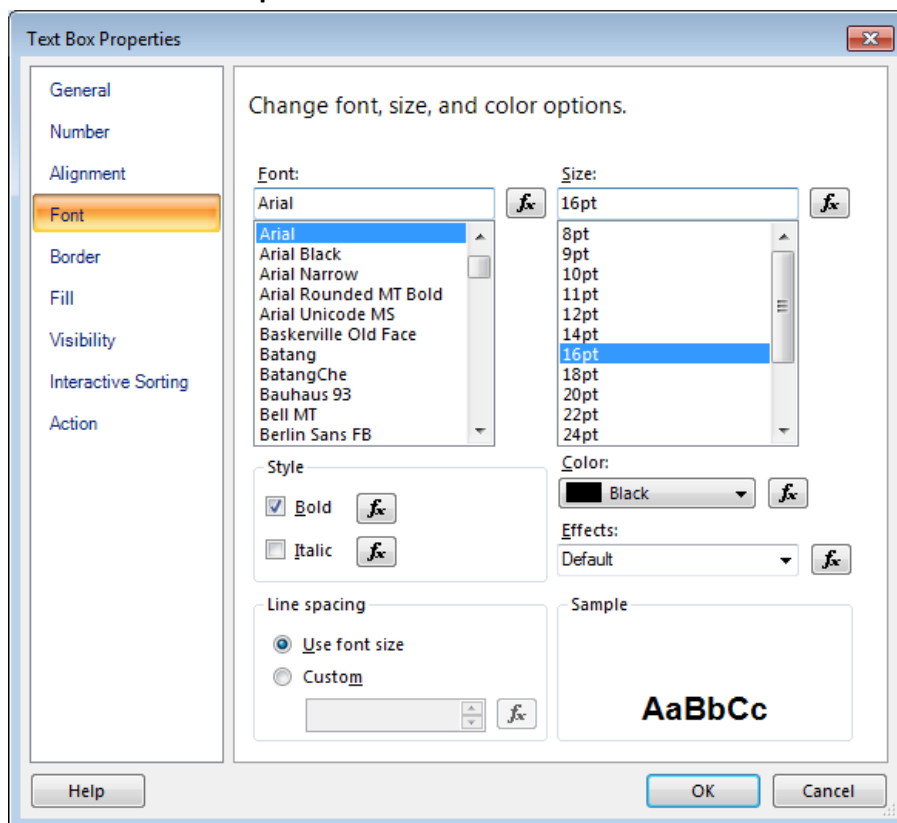
1. Right click the mouse off the visible report page (i.e. Above, Below, Left or Right) and select **Add Page Header** from the popup menu
2. Right click in the **Page Header** and select **Header Properties ...** from the popup menu.
3. Select **General** in the **Page Header Properties** dialog
 - a. Enter 2.5cm for the **Height**.




4. Click **OK** to accept the changes and close the **Page Header Properties** dialog
5. Right click in the **Page Header** and select **Insert** then **Text Box** from the popup menu.
6. Right click in the **Text Box** and select **Expression ...** from the popup menu
 - a. Select **Variables** under the **Categories** list and double click the **report_title** item in the **Values** list.



7. Click **OK** to accept the changes and close the **Expression** dialog
8. Right click the new **Text Box** and select **Text Box Properties ...** from the popup menu.
9. Select **Font** in the **Text Box Properties** dialog
 - a. Select the **16pt** from the **Size** list and check the box beside **Bold** under **Style**.



10. Click **OK** to accept the changes and close the **Text Box Properties** dialog
11. Click the Text <<Expr>>. When correctly selected the **Properties** view will show **Selected Text**.
12. In the **Properties** view, select the **Categorized** icon and locate the **General** category.
13. Enter `Report_Title` for the **Label** value
14. Position the text box close to the left side and resize the Text Box to allow the text to be visible and make it the width of the Table.
15. Right click in the **Page Header** and select **Insert** then **Text Box** from the popup menu.
16. Right click in the **Text Box** and select **Expression ...** from the popup menu
 - a. Select the text in the **Set expression for: Value** field and press the **Delete** key to remove all text
 - b. Paste the following text into the **Set expression for: Value** field:


```
="Report Interval: " & Parameters!report_interval.Value
```
17. Click **OK** to accept the changes and close the **Expression** dialog
18. Click the Text <<Expr>>. When correctly selected the **Properties** view will show **Selected Text**.
19. In the **Properties** view, select the **Categorized** icon and locate the **General** category.
20. Enter `Report_Interval` for the **Label** value
21. Position the text box close to the left side and resize the Text Box to allow the text to be visible and make it the width of the Table
22. Test using report viewer.
23.  Save the report.

6.4.3.4. Add Report Footer

New Reports do contain a Report Footer; however the default size will not work for this example. The Report Footer height will be increased. The items included in this example are the runtime directives:

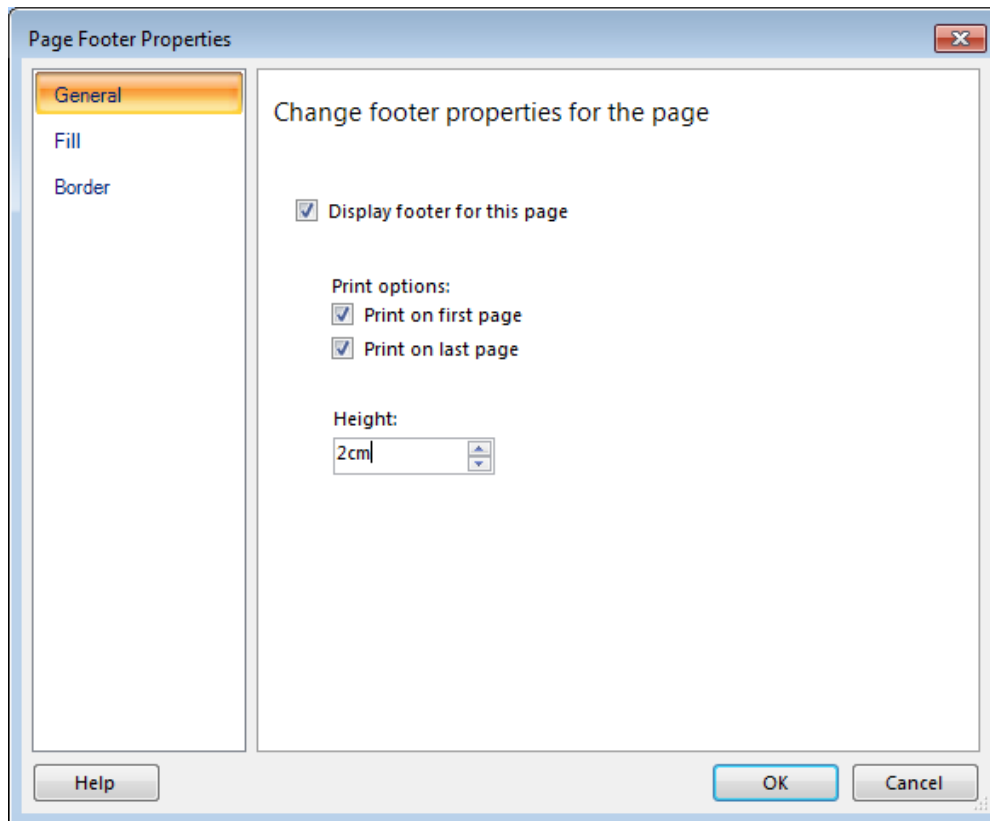
- `report_filepath`
- `report_user`

6.4.3.4.1. Before you begin

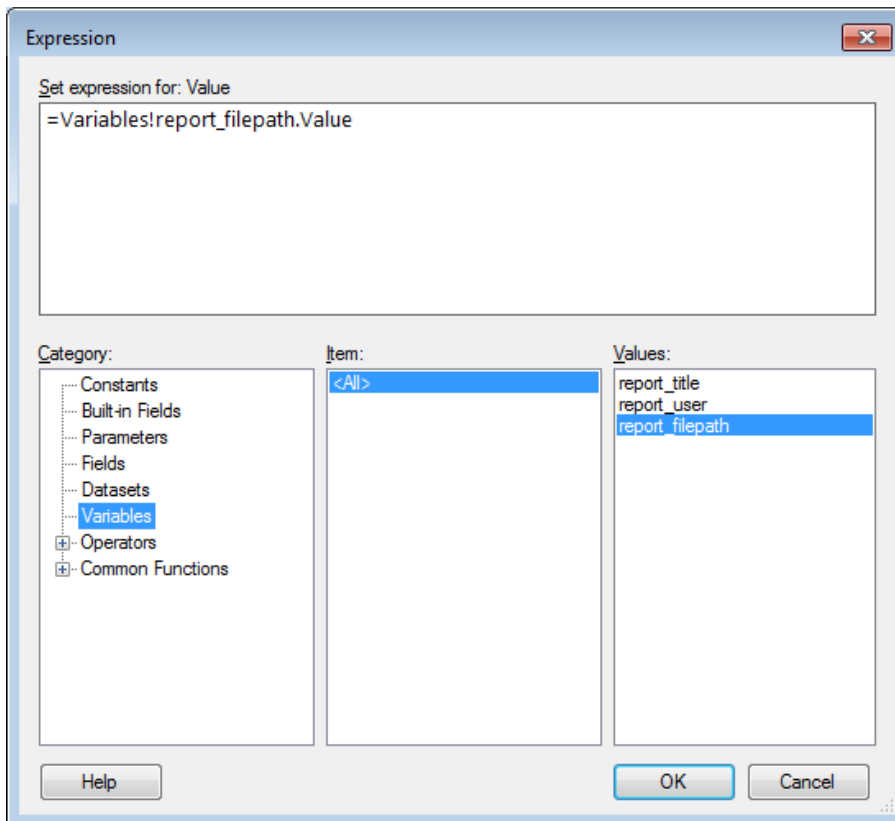
Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

6.4.3.4.2. Procedure

1. Right click in the **Page Footer** and select **Footer Properties ...** from the popup menu.
2. Select **General** in the **Page Footer Properties** dialog
3. Enter `2cm` for the **Height**.



4. Click **OK** to accept the changes and close the **Page Footer Properties** dialog
5. Right click in the **Page Footer** and select **Insert** then **Text Box** from the popup menu.
6. Right click in the **Text Box** and select **Expression ...** from the popup menu
7. Select **Variables** under the **Categories** list and double click the **report_filepath** item in the **Values** list.



8. Click **OK** to accept the changes and close the **Expression** dialog
9. Right click the new **Text Box** and select **Text Box Properties ...** from the popup menu.
10. Select **Font** in the **Text Box Properties** dialog
11. Enter **7pt** for the **Size**. This must be typed in since 8 pt is the smallest size available to select.
12. Click the Text **<<Expr>>**. When correctly selected the **Properties** view will show **Selected Text**.
13. In the **Properties** view, select the **Categorized** icon and locate the **General** category.
14. Enter `Report_FilePath` for the **Label** value
15. Position the text box close to the left side of the report and resize the Text Box to allow the text to be visible and make it the width of the Table.
16. Right click in the **Page Footer** and select **Insert** then **Text Box** from the popup menu.
17. Right click in the **Text Box** and select **Expression ...** from the popup menu
18. Select the text in the **Set expression for: Value** field and press the **Delete** key to remove all text
19. Paste the following text into the **Set expression for: Value** field:


```
="Printed by: " & Variables!report_user.Value & " on " &
      Globals!ExecutionTime
```
20. Click **OK** to accept the changes and close the **Expression** dialog
21. Click the Text **<<Expr>>**. When correctly selected the **Properties** view will show **Selected Text**.
22. In the **Properties** view, select the **Categorized** icon and locate the **General** category.
23. Enter `PrintedByOn` for the **Label** value

24. Position the text box close to the left side and resize the Text Box to allow the text to be visible and make about half the width of the Table
25. Right click in the **Page Footer** beside the previous text box and select **Insert** then **Text Box** from the popup menu.
26. Right click in the **Text Box** and select **Expression ...** from the popup menu
27. Select **Built-in Fields** under the **Categories** list and double click the **OverallPageNumber** item in the **Values** list.
28. Click **OK** to accept the changes and close the **Expression** dialog
29. Click the Text <<Expr>>. When correctly selected the **Properties** view will show **Selected Text**.
30. In the **Properties** view, select the **Categorized** icon and locate the **General** category.
31. Enter `PageNumber` for the **Label** value
32. Position the text box close to the right side and it is aligned with the previous text box. Resize the Text Box to allow the text to be visible, and not overlapping the previous text box.
33. Test using report viewer.
34. Save the report.

6.4.4. Report Settings

6.4.4.1. Page Setup

Page Setup configures the size, orientation and margins reserved for the report. The values are used when exporting and printing the report.

Items on the report should not run to the edge of the page. Reposition and resize the report items to allow some white space around the edge of the report.

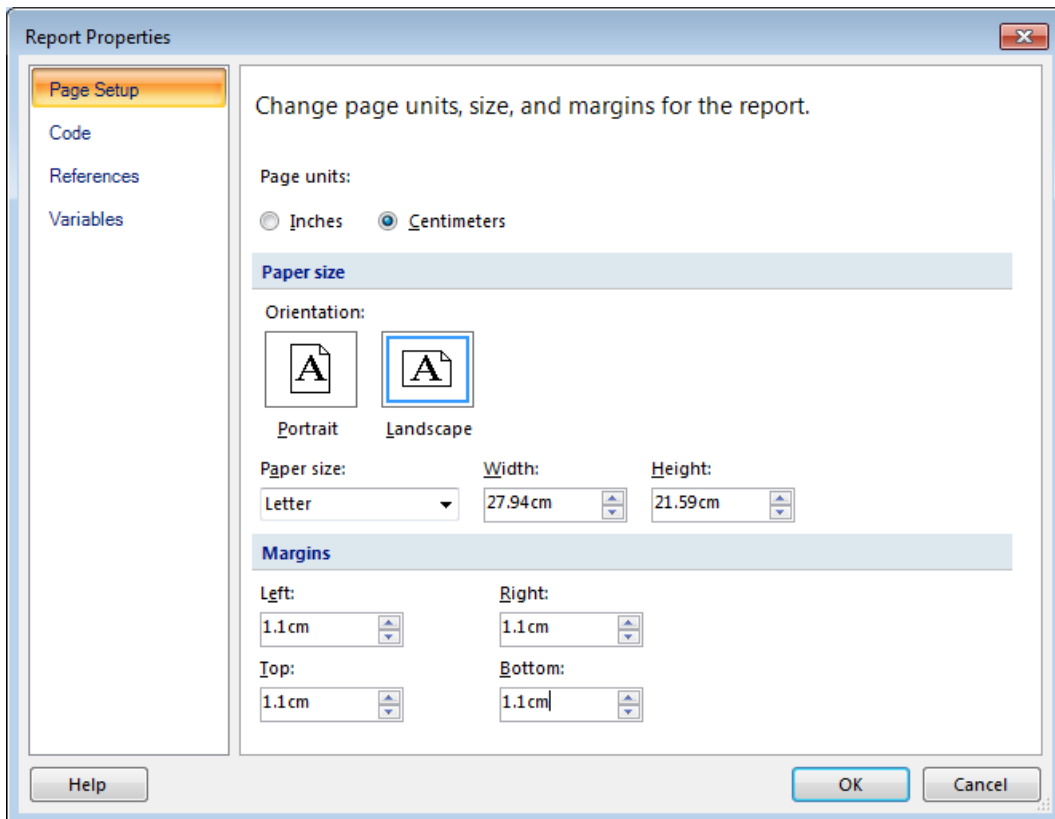
This procedure uses settings that have been predefined. For details on how these values were selected, see [Page Size](#) and [Margins](#) in [Appendix C: Page Setup](#).

6.4.4.1.1. Before you begin

Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

6.4.4.1.2. Procedure

1. Right click the mouse off the visible report page (i.e. Above, Below, Left or Right) and select **Report Properties ...** from the popup menu
2. Select **Page Setup** in the **Report Properties** dialog
 - a. Select **Landscape** under **Orientation**
 - b. Select **Letter** under **Page Size**
 - c. Enter `1.1cm` for the **Left, Right, Top** and **Bottom** under **Margins**



3. Click **OK** to accept the changes and close the **Report Properties** dialog
4. Test using report viewer.
5. Save the report. Suggested name
AgentPerformance_ReportSettings.rdl

6.4.4.2. Page Settings

The interactive size represents the size shown to the user when viewed in the Report Viewer. The interactive size determines the maximum size of the report body per page. This size is reduced by the size of the following report items:

- Margins
- Header Height
- Footer Height

Setting the background color of the report ensures the color is fixed. The default setting treats the background as transparent allowing the color of the Report Viewer screen to control the color.

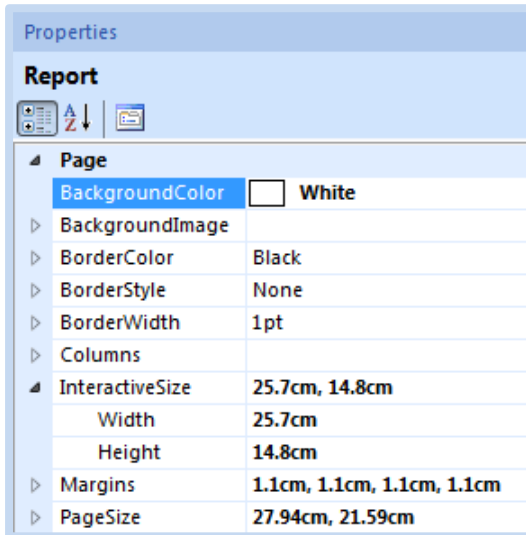
This procedure uses settings that have been predefined. For details on how these values were selected, see [Interactive Size](#) in [Appendix C: Page Setup](#).

6.4.4.2.1. Before you begin

Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

6.4.4.2.2. Procedure

1. Select the Report by clicking the mouse off the visible report page (i.e. Above, Below, Left or Right). When correctly selected the **Properties** view will show **Report** as the selected item
2. In the **Properties** view, select the **Categorized** icon and locate the **Page** category.
3. Expand the **InteractiveSize** settings and enter the following values:
 - a. Width: 25.7cm
 - b. Height: 14.8cm
4. Click the drop down list beside **BackgroundColor** and click the **White** color tile. Hover the mouse over the tiles to display the color name.



5. Test using report viewer, print layout and export.
6. Save the report.

6.4.4.3. Report Body Settings

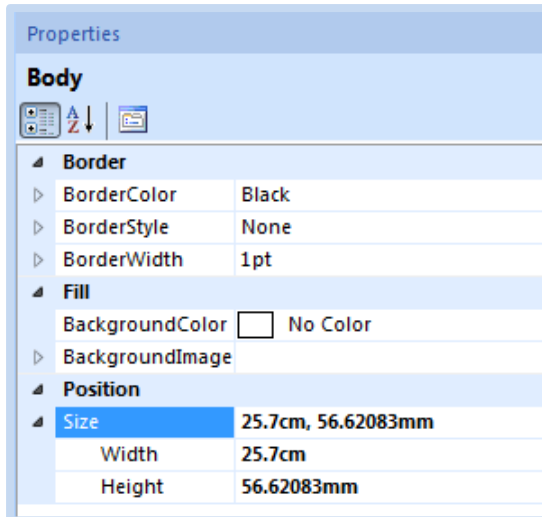
The Report Body height and width automatically adjusts when objects are moved and resized while designing the report. Now that the interactive size has been set, the body width can help show where the effective width ends.

6.4.4.3.1. Before you begin

Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

6.4.4.3.2. Procedure

1. Select the Report Body by clicking the mouse between the Header and Footer, and not clicking a report item like the table. When correctly selected the **Properties** view will show **Body** as the selected item
2. In the **Properties** view, select the **Categorized** icon and locate the **Position** category.
3. Expand the **Size** settings and enter the following value:
 - a. Width: 25.7cm



4. Resize the report body Height by clicking and dragging the line in the vertical ruler where the Footer section starts.
5. Resize and reposition other objects as required.
6. Test using report viewer.
7. Save the report.

6.4.4.4. Localization Settings

ACCS automatically updates the Language setting under Localization to match the preferred language of the user. The preferred language is the first language set in the web browser.

While developing a report, set the language to the same settings as the expected users of the report, and ensure the format of values are displayed correctly.

The report language is set to the Culture Name, which is typically represented by a “2 character Language – 2 character Country”. In the browser the culture name should be visible along with the full name. The following table shows a few of these values for reference:

Culture Name	Language
en-US	English (United States)
en-UK	English (United Kingdom)
de-DE	German (Germany)
fr-FR	French (France)
fr-CA	French (Canada)
es-ES	Spanish (Spain)
it-IT	Italian (Italy)
ja-JP	Japanese (Japan)
pt-BR	Portuguese (Brazil)
ko-KR	Korean (Korea)
ru-RU	Russian (Russia)
zh-CN	Chinese (People's Republic of China)
zh-TW	Chinese (Taiwan)

For a list of Culture Names, see:

<http://msdn.microsoft.com/en-us/goglobal/bb896001.aspx>

6.4.4.4.1. Before you begin

Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

6.4.4.4.2. Procedure

1. Select the Report by clicking the mouse off the visible report page (i.e. Above, Below, Left or Right). When correctly selected the **Properties** view will show **Report** as the selected item
2. In the **Properties** view, select the **Categorized** icon and locate the **Localization** category.
3. Click the drop down list beside **Language** and select the user Locale. For US English the value is **en-US**.
4. Test using report viewer.
5. Save the report.

6.4.5. Document Map

A document map provides a set of navigational links to report items. A user can click links in the document map to jump to the report page that displays that item.

How the document map appears is based on the target:

Target	Document Map
Report Viewer	Navigation Tree
PDF	Bookmarks
Excel	Worksheet named Document Map with Hyperlinks
Word	Table of contents with Hyperlinks
Print	Does not print

6.4.5.1. Limitations

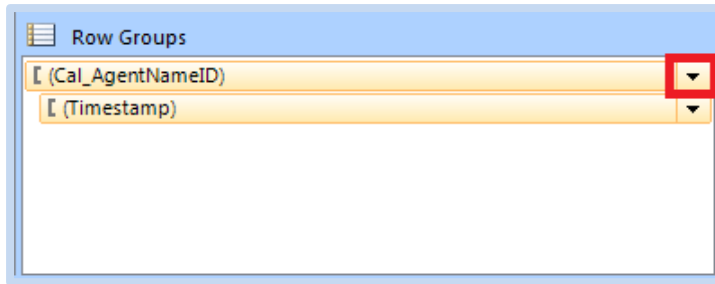
As the number of items in the document increases, the time to generate the report and navigate pages increases. This example does not limit the document map size and includes the document when exporting to Excel and Word. For details on adding these settings, see [Appendix D: Document Map](#).

6.4.5.2. Before you begin

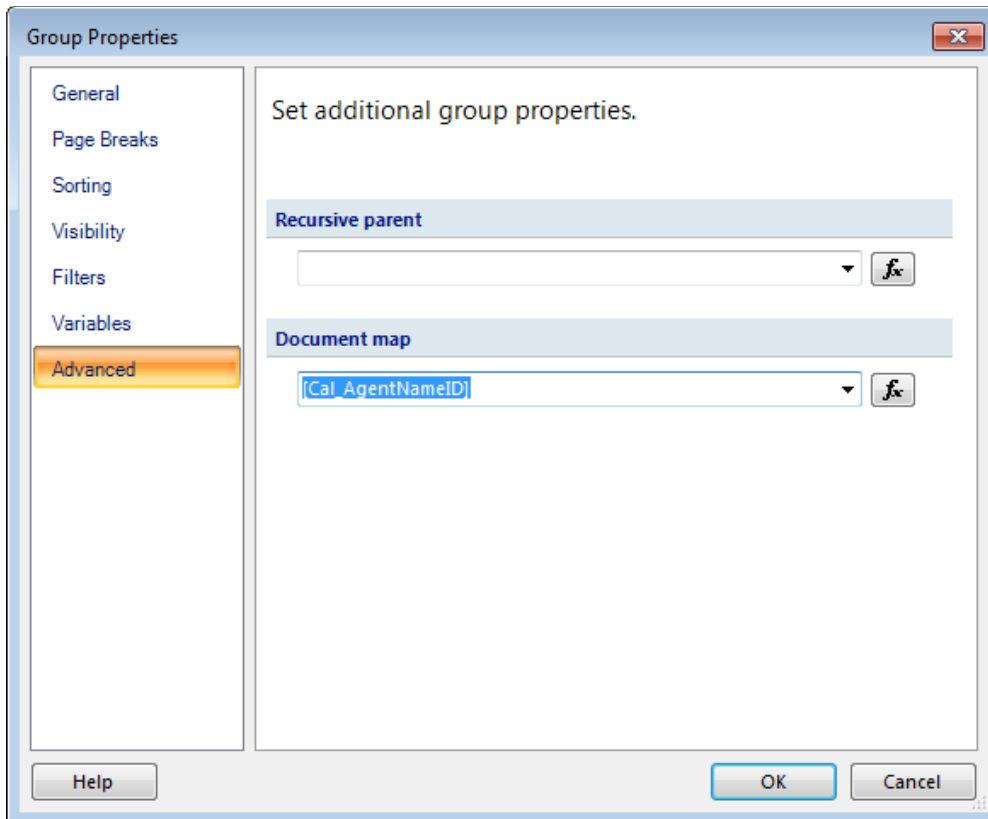
Make sure the **Grouping** view is visible. If **Grouping** is not visible select the **View** menu item and check the box for **Grouping**.

6.4.5.3. Procedure

1. Click the arrow beside the **(Cal_AgentNameID)** group in the **Row Groups** list and select **Group Properties** from the popup menu.



2. Select **Advanced** in the **Group Properties** dialog.
3. Select **[Cal_AgentNameID]** from the drop down list under **Document Map**.



4. Click **OK** to accept the changes and close the **Group Properties** dialog
5. Test using report viewer and export.
6. Save the report. Suggested name `AgentPerformance_DocumentMap.rdl`

6.5. ACCS Integration

The steps in this section are required to integrate the report with ACCS Historical Reporting. If the report is not going to use this functionality, this section can be skipped.

For further details, see [Appendix E: ACCS Integration](#).

6.5.1. Runtime Variables and Parameters

This example has already integrated some of the runtime variables and parameters.



For details, see [Add Runtime Report Variables](#) and [Add Runtime Report Parameters](#) in the [Header and Footer](#) section of this example.

6.5.2. Runtime Query Parameters

Proceed with the tasks in this section if:

- Report will be filtered based on the user selected Data Range
- Report will be filtered based on the user's Data Partitioning

6.5.2.1. Gather required details

The details gathered from the Dataset will be used in the creation of the runtime query parameters.

6.5.2.1.1. Before you begin

The existing Dataset should be defined and tested.

6.5.2.1.2. Procedure

1. Append the Dataset name to the runtime query parameter names:

Parameter	Name
Timestamp	report_timestamp_dAgentPerformanceStat
Row Filter	report_rowfilter_dAgentPerformanceStat
Query	report_query_dAgentPerformanceStat

2. Convert the existing Dataset query to a query parameter value
 - a. Copy the text of the existing Dataset query and paste it into a text editor application (i.e. Notepad). The existing query is:

```
SELECT TOP 100
MIN(dAgentPerformanceStat.Timestamp) AS Timestamp,
MIN(LTRIM(dAgentPerformanceStat.AgentLogin)) AS
AgentLogin,
MIN(LTRIM(dAgentPerformanceStat.AgentSurName)) AS
AgentSurName,
MIN(LTRIM(dAgentPerformanceStat.AgentGivenName)) AS
AgentGivenName,
SUM(dAgentPerformanceStat.CallsOffered) AS
CallsOffered,
SUM(dAgentPerformanceStat.CallsAnswered) AS
CallsAnswered,
SUM(dAgentPerformanceStat.LoggedInTime) AS
LoggedInTime,
SUM(dAgentPerformanceStat.TalkTime) AS TalkTime,
SUM(dAgentPerformanceStat.HoldTime) AS HoldTime,
SUM(dAgentPerformanceStat.PostCallProcessingTime) AS
PostCallProcessingTime,
```

```
SUM(dAgentPerformanceStat.NotReadyTime) AS
NotReadyTime,
SUM(dAgentPerformanceStat.WaitingTime) AS WaitingTime
FROM dbo.dAgentPerformanceStat dAgentPerformanceStat
WHERE (dAgentPerformanceStat.Timestamp>= DateAdd('d', -
8, Now()) And dAgentPerformanceStat.Timestamp<=
DateAdd('d', -1, Now()))
GROUP BY dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin
ORDER BY dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin
```

- b. The report is grouping the data, which added a default ascending sort. Remove the following ORDER BY clause:

```
ORDER BY dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin
```

- c. Remove the following items from the query, which are filters for design purposes only:

- i. TOP 100 in the SELECT clause
- ii. (dAgentPerformanceStat.Timestamp>= DateAdd('d', -8, Now()) And dAgentPerformanceStat.Timestamp<= DateAdd('d', -1, Now())) in the WHERE clause

- d. There are now no remaining conditions in the WHERE clause, so the Available Value for the row filter parameter will be:

Label	Value
Default	Default

- e. Replace the WHERE clause with the formatted row filter name:
{?report_rowfilter_dAgentPerformanceStat}

The query produced from this process is:

```
SELECT
MIN(dAgentPerformanceStat.Timestamp) AS Timestamp,
MIN(LTRIM(dAgentPerformanceStat.AgentLogin)) AS
AgentLogin,
MIN(LTRIM(dAgentPerformanceStat.AgentSurName)) AS
AgentSurName,
MIN(LTRIM(dAgentPerformanceStat.AgentGivenName)) AS
AgentGivenName,
```

```

SUM(dAgentPerformanceStat.CallsOffered) AS
CallsOffered,
SUM(dAgentPerformanceStat.CallsAnswered) AS
CallsAnswered,
SUM(dAgentPerformanceStat.LoggedInTime) AS
LoggedInTime,
SUM(dAgentPerformanceStat.TalkTime) AS TalkTime,
SUM(dAgentPerformanceStat.HoldTime) AS HoldTime,
SUM(dAgentPerformanceStat.PostCallProcessingTime) AS
PostCallProcessingTime,
SUM(dAgentPerformanceStat.NotReadyTime) AS
NotReadyTime,
SUM(dAgentPerformanceStat.WaitingTime) AS
WaitingTime
FROM dbo.dAgentPerformanceStat dAgentPerformanceStat
{?report_rowfilter_dAgentPerformanceStat}
GROUP BY dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin

```

6.5.2.2. *Timestamp Parameter*

Proceed with this task if:

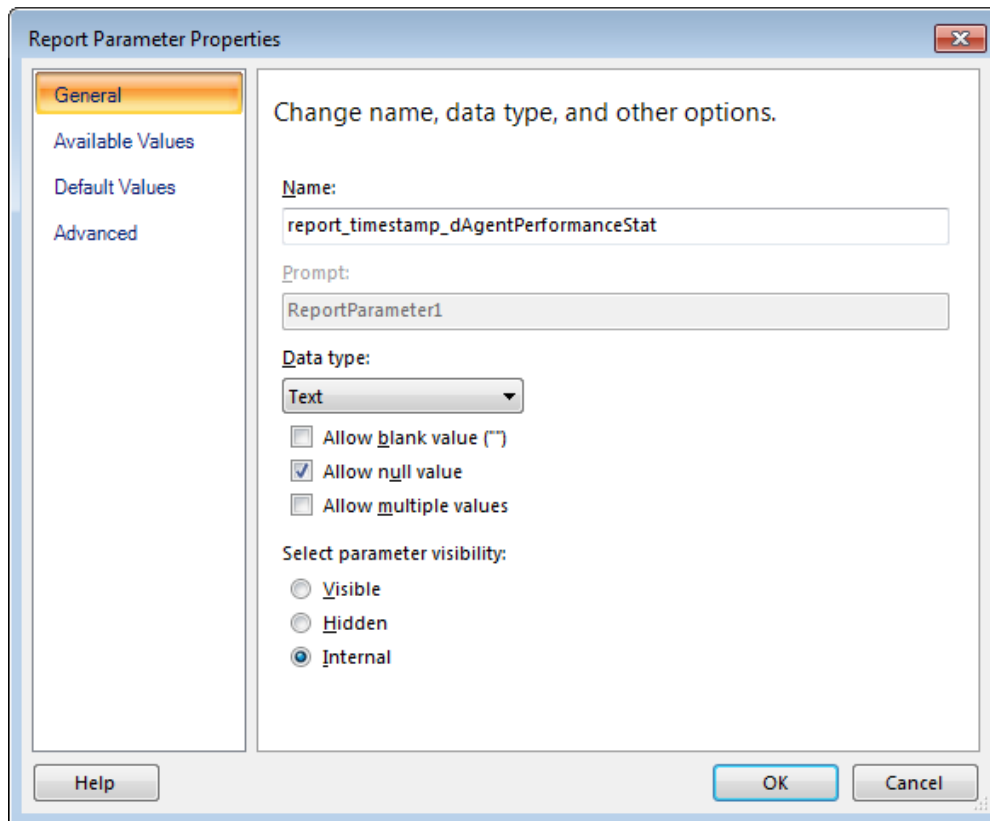
- Using Data Range filtering and system is ACCS 6.4 SP13
- Data Range filtering should be applied to multiple fields

6.5.2.2.1. *Before you begin*

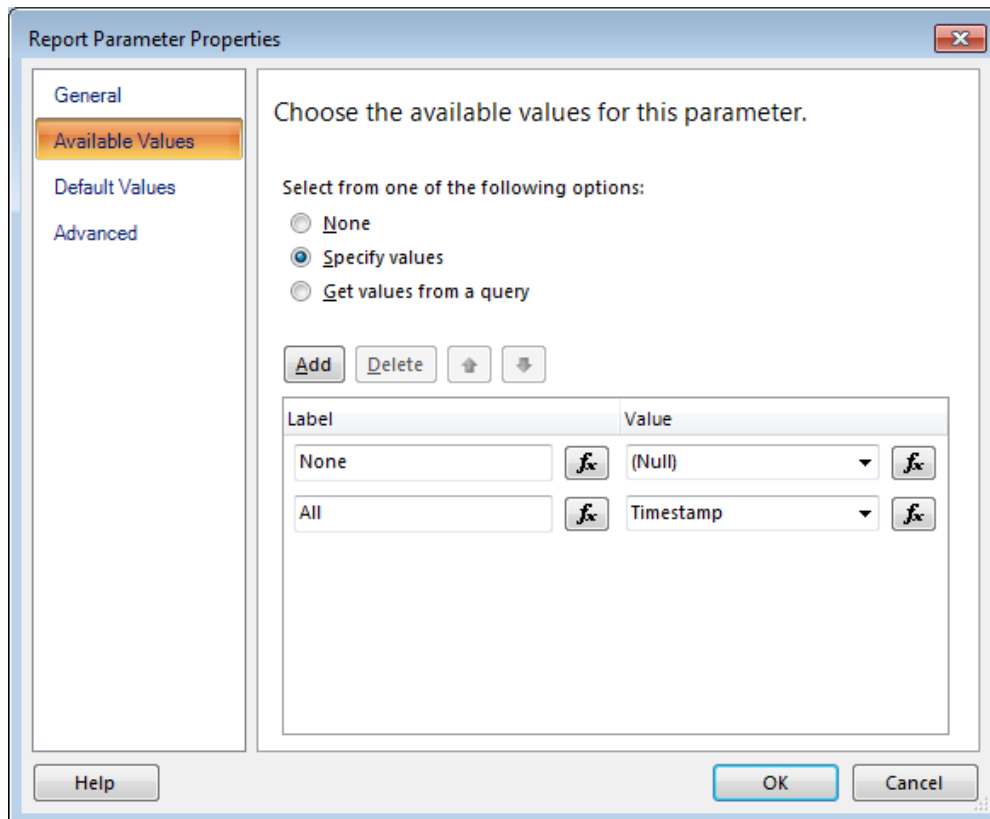
Make sure that the **Report Data** tree view is visible. If **Report Data** is not visible select the **View** menu item and check the box for **Report Data**.

6.5.2.2.2. *Procedure*

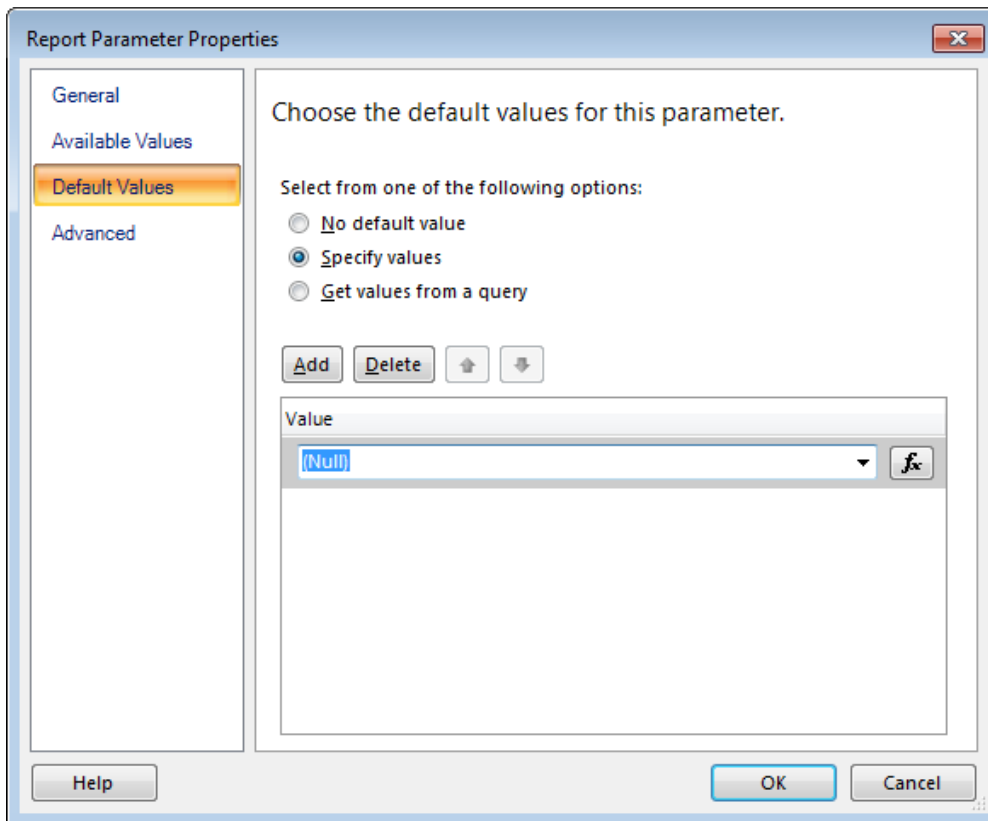
1. Right click **Parameters** in the **Report Data** tree view
2. Select **Add Parameter ...** from the pop up menu
3. Select **General** in the **Report Parameter Properties** dialog
 - a. Enter `report_timestamp_dAgentPerformanceStat` in the **Name** field.
 - b. Select **Text** for **Data Type** from the drop down list
 - c. Check the **Allow null value** check box
 - d. Select **Internal** under **Select parameter visibility**



4. Select **Available Values** in the **Report Parameter Properties** dialog
 - a. Select **Specify values** under **Select from one of the following options**
 - b. Click **ADD** and enter the following:
 - i. Label: None
 - ii. Value: Select (Null) from the drop down list
 - c. Click **ADD** and enter and enter the following:
 - i. Label: All
 - ii. Value: Timestamp



5. Select **Default Values** in the **Report Parameter Properties** dialog
 - a. Select **Specify values** under **Select from one of the following options**
 - b. Click **ADD** and enter the following:
 - i. Value: Select (Null) from the drop down list



6. Click **OK** to accept the changes and close the **Report Parameter Properties** dialog
7. Test using report viewer.
8. Save the report. Suggested name `AgentPerformance.rdl`

6.5.2.3. Row Filter Parameter

The row filter parameter provides a reference in the query where filtering is applied. This parameter must be defined if the report is:

- Report will be filtered based on the user selected Data Range
- Report will be filtered based on the user's Data Partitioning

6.5.2.3.1. Before you begin

Make sure that the **Report Data** tree view is visible. If **Report Data** is not visible select the **View** menu item and check the box for **Report Data**.

6.5.2.3.2. Procedure

1. Right click **Parameters** in the **Report Data** tree view and select **Add Parameter ...** from the pop up menu
2. Select **General** in the **Report Parameter Properties** dialog
 - a. Enter `report_rowfilter_dAgentPerformanceStat` in the **Name** field
 - b. Select **Text** for **Data Type** from the drop down list
 - c. Check the **Allow null value** check box
 - d. Select **Internal** under **Select parameter visibility**

3. Select **Available Values** in the **Report Parameter Properties** dialog
 - a. Select **Specify values** under **Select from one of the following options**
 - b. Click **ADD** and enter the following:
 - i. **Label:** Default
 - ii. **Value:** Default
 - c. Click **ADD** and enter the following:
 - i. **Label:** TableAlias
 - ii. **Value:** *
 - d. Click **ADD** and enter the following:
 - i. **Label:** FieldName
 - ii. **Value:** *
4. Select **Default Values** in the **Report Parameter Properties** dialog
5. Select **Specify values** under **Select from one of the following options**
6. Click **ADD** and enter the following:
 - a. **Value:** Select `Default` from the drop down list
7. Click **OK** to accept the changes and close the **Report Parameter Properties** dialog
8. Test using report viewer.

6.5.2.4. Query Parameter

A query parameter allows the report to have a different runtime query than the original design time Dataset query. This parameter must be defined if the report is:

- Report will be filtered based on the user selected Data Range
- Report will be filtered based on the user's Data Partitioning

6.5.2.4.1. Before you begin

Make sure that the **Report Data** tree view is visible. If **Report Data** is not visible select the **View** menu item and check the box for **Report Data**.

6.5.2.4.2. Procedure

1. Right click **Parameters** in the **Report Data** tree view and select **Add Parameter ...** from the pop up menu
2. Select **General** in the **Report Parameter Properties** dialog
 - a. Enter `report_query_dAgentPerformanceStat` in the **Name** field
 - b. Select **Text** for **Data Type** from the drop down list
 - c. Check the **Allow null value** check box
 - d. Select **Internal** under **Select parameter visibility**
3. Select **Available Values** in the **Report Parameter Properties** dialog
 - a. Select **Specify values** under **Select from one of the following options**
 - b. Click **ADD** and enter the following:
 - i. **Label:** Default
 - ii. **Value:** Default
 - c. Click **ADD** and enter the following:

- i. **Label:** All
- ii. Click the **fx** button beside the empty **Value** field
- iii. Paste the following text into the **Set expression for: Value** field:

```
SELECT
MIN(dAgentPerformanceStat.Timestamp) AS Timestamp,
MIN(LTRIM(dAgentPerformanceStat.AgentLogin)) AS
AgentLogin,
MIN(LTRIM(dAgentPerformanceStat.AgentSurName)) AS
AgentSurName,
MIN(LTRIM(dAgentPerformanceStat.AgentGivenName)) AS
AgentGivenName,
SUM(dAgentPerformanceStat.CallsOffered) AS
CallsOffered,
SUM(dAgentPerformanceStat.CallsAnswered) AS
CallsAnswered,
SUM(dAgentPerformanceStat.LoggedInTime) AS
LoggedInTime,
SUM(dAgentPerformanceStat.TalkTime) AS TalkTime,
SUM(dAgentPerformanceStat.HoldTime) AS HoldTime,
SUM(dAgentPerformanceStat.PostCallProcessingTime) AS
PostCallProcessingTime,
SUM(dAgentPerformanceStat.NotReadyTime) AS
NotReadyTime,
SUM(dAgentPerformanceStat.WaitingTime) AS
WaitingTime
FROM dbo.dAgentPerformanceStat dAgentPerformanceStat
{?report_rowfilter_dAgentPerformanceStat}
GROUP BY dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin
```

- iv. Click **OK** to accept the changes and close the **Expression** dialog
4. Select **Default Values** in the **Report Parameter Properties** dialog
 - a. Select **Specify values** under **Select from one of the following options**
 - b. Click **ADD** and enter the following:
 - i. **Value:** Select `Default` from the drop down list
5. Click **OK** to accept the changes and close the **Report Parameter Properties** dialog
6. Test using report viewer.
7. Save the report.

6.6. Import

When a report is imported, the report is saved on the Contact Center server. The report is accessed by opening the Private Report Templates folder.



To share the report with other users, a copy of the report must be saved in a Group Folder.

For further details, see [Appendix F: Import](#).

6.6.1.1. Before you begin

Log on to CCMA and launch Historical Reporting.

6.6.1.2. Procedure

1. In the left pane, click the server to which to import the new report template.
2. On the **Report** menu, click **Import**.
3. The system displays the **Template Importing Wizard** dialog box.
4. Click the **Browse** button next to the **Select Your Report template** field, and navigate to the report.
5. In the **Template Name** box, type a unique name for the report.
6. From the **Report is based on the following data** list, select **Historical**.
7. For Database alias and Timestamp field enter:

```
dAgentPerformanceStat . Timestamp
```

8. From the **Report data range** list, select **Daily**.
9. Click **Import**.

7. Appendix A: ACCS Databases

It's important to have some understanding of the ACCS database schemas. The Avaya Aura Contact Center Performance Management Data Dictionary is a detailed reference guide to the database schemas.

The ACCS solution stores data related to reporting in either the CCMS or CCMM database. Reports are created using the views and do not directly access the tables.

7.1. CCMS Views

The CCMS database views offer several different types of data. The types are defined as:

- Summarized Historical Data
- Contact Statistics
- Event Statistics
- Configuration

Some of the views available from the CCMS database are not supported for ACCS. The views shown in this section are a subset of the available views, specific to ACCS.

7.1.1. Qualified name

The qualifier for the CCMS views is "dbo.", which is also the default qualifier if one is not provided. This means that the qualifier does not need to be included when referencing the CCMS views.

For example, the qualified name for the dAgentPerformanceStat view is:

```
dbo.dAgentPerformanceStat
```

Using the view name dAgentPerformanceStat will also reference the same view.

7.1.2. Summarized Historical Statistics

The CCMS database defines rolled up statistics for Interval, Daily, Weekly and Monthly. These are referred to as Summarized historical statistics. The rolled up versions of these differ only by the starting character.

The following table shows the difference using the Agent Performance Statistics view as an example:

Rolled up to	Starting character	Example
Interval (15 mins)	i	iAgentPerformanceStat
Daily	d	dAgentPerformanceStat

Rolled up to	Starting character	Example
Weekly	w	wAgentPerformanceStat
Monthly	m	mAgentPerformanceStat

7.1.3. Call By Call Statistics

Call by Call Statistics are event based and differ from the other views because the statistics are provided in specific date views.

The format of the view name is eCallByCallStatYYYYMMDD where YYYYMMDD is replaced by the date. For example, to view the Call By Call Statistics for November 17th 201, the view would be eCallByCall20141117.

If adding a Dataset to a report based on Call by Call statistics, the Dataset name must start with eCallByCallStat.

7.2. CCMM Views

The available views within the CCMM database are defined to offer statistics for contacts related to:

- Multimedia
- Outbound

7.2.1. Qualified name

The qualifier for the CCMM views used for reporting is “vw.”, which is not a default qualifier. This means that the qualifier must be included when referencing the CCMM views.

For example, the qualified name for the ContactsOutstandingDetail view is:

```
vw.ContactsOutstandingDetail
```

Using the view name ContactsOutstandingDetail in the SQL Select statement without the qualifier will fail.

7.3. Primary Keys

The views define one or more primary keys that identify a record as unique. With the exception of views showing different rolled up intervals of the same data, each database view has a different set of keys to identify a unique record.

For each of the CCMS view types, a table is provided that indicates which database fields are primary keys.

7.3.1. Duplicate Records on Reports

When building a report, it is important to know what makes a record unique. If the report does not show all of the keys that make a record unique, the report can appear to show duplicate records, since the key that would identify the record as unique is not shown.

To ensure this does not happen, the following options are available:

- Show all primary keys on the report
- Use `Group By` in the query with the appropriate aggregate on each field
- Use groups on the report with the appropriate aggregate on each field, and hide any sections that are not part of a group.

7.3.2. CCMS Summarized Historical Statistics

The following table shows the primary keys for the CCMS Summarized Historical Statistics views and only shows the interval version; however the same primary keys apply to the Daily, Weekly and Monthly views:

View	Timestamp	UserID / AgentLogin	SupervisorUserID / SupervisorLogin	ApplicationID	SkillsetID	ActivityCode	CDN	DNIS	RouteID
iActivityCodeStat	✓	✓		✓		✓			
iAgentByApplicationStat	✓	✓		✓					
iAgentBySkillsetStat	✓	✓			✓				
iAgentPerformanceStat	✓	✓	✓						
iApplicationStat	✓			✓					
iCDNStat	✓						✓		
iDNISStat	✓							✓	
iRANMusicRouteStat	✓								✓
iRouteStat	✓								✓
iSkillsetStat	✓			✓	✓				

7.3.2.1. Agent and Supervisor

For CCMS Summarized Historical Statistics views, the unique Agent is defined by the primary key UserID, and when relevant, the unique Supervisor is defined by the primary key SupervisorUserID.

The ACCS standard reports use the AgentLogin and SupervisorLogin as if they were primary keys. The reasons for doing this include:

- The UserID fields are generated within CCMS, and not known by users
- If an Agent or Supervisor is accidentally deleted, when they are added to the system again, a new UserID is created.

7.3.3. CCMS Contact Statistics

The following table shows the primary keys for the CCMS Contact Statistics views:

View	ContactOriginatedStamp	OriginatedStamp	CCMID	CodeStartStamp	Code	ObserveBargeln StartStamp	ObserveBargeln Type	IntrinsicKey
eCSRCodeStat	✓	✓	✓	✓	✓			
eCSRIntrinsicStat	✓	✓	✓					✓
eCSRObserveBargelnStat	✓	✓	✓			✓	✓	
eCSRStat	✓	✓	✓					

7.3.4. CCMS Event statistics

The following table shows the primary keys for the CCMS Event Statistics views:

View	Timestamp	UserID	AgentLogin	TelsetLoginID	EventType	CallID	CallEvent
eAgentLoginStat	✓	✓	✓		✓		
eCallByCallStatYYYYMMDD	✓			✓		✓	✓

7.3.5. CCMS Configuration

The following table shows the primary keys for the CCMS Configuration views:

Name	Primary Keys
AccessRights	PCLoginName
ActivityCode	ActivityCode
ActivityCodeByContactType	ActivityCode ContactID
ActivityCodeBySkillset	ActivityCode SkillsetID
Agent	UserID / TelsetLoginID
Application	ApplicationID
ApplicationByScript	ParentName
ApplicationTemplate	TemplateID Field
ApplicationThresholdTemplate	TemplateID Field
CDN	CDN
CodeToMessageMap	Category Code
ContactType	ContactID
DNIS	DNIS
DNISThresholdTemplate	TemplateID
Formula	FormulaID
HistoricalStatCollection	
HistoricalStatDuration	
HistoricalStatStorage	Parameter
Provider	ProviderID
Route	RouteID
RouteThresholdTemplate	TemplateID FieldID
ScheduledSkillsetAssignment	AssignID SkillsetID UserID / UserTelsetLoginID
ScheduledSupervisorAssignment	AssignID SupervisorID AgentID
Script	ScriptID
ScriptVariableProperties	Name
ScriptVariables	Script Variable
Site	SiteID
Skillset	SkillsetID
SkillsetByAgent	SkillsetID UserID
SkillsetByAssignment	AssignID SkillsetID UserID / UserTelsetLoginID

Name	Primary Keys
SkillsetThresholdTemplate	TemplateID FieldID
SummaryThresholdTemplate	TemplateID FieldID
Supervisor	UserID / TelsetLoginID
SupervisorAgentAssignment	SupervisorUserID / TelsetLoginID AgentUserID / AgentTelsetLoginID
SupervisorByAssignment	AssignID SupervisorID AgentID
Treatment	TreatmentID
URIByAgent	UserID
UserTemplate	UserThresholdTemplate
UserThresholdTemplate	ThresholdTemplateID FieldID
UserValidationReport	sTimestamp UserID / TelsetLoginID InvalidCode
UserValidationStatus	sTimestamp UserID / TelsetLoginID InvalidCode

7.4. Data Partitioning

ACCS allows a user to be defined with specific Data Partitioning. The partition elements used by Historical Reporting are:

- Agent
- Application
- CDN
- DNIS
- Skillset

7.4.1. Using Data Partitioning

Historical Reporting applies the data partitioning defined for a user based on the database view name. For a report to use the data partitioning, the report must use the view name as the Dataset name.

The easiest way to accomplish this is to name the Dataset using the same name as the unqualified database view name defined in the FROM clause of the query used to create the Dataset.

For example, if the SQL Select statement FROM clause is:

```
FROM dbo.dAgentPerformanceStat
```



Then the name to use for the Dataset would be dAgentPerformanceStat without the qualifier “dbo.”.

If the SQL Select statement FROM clause uses an alias:

```
FROM dbo.dAgentPerformanceStat iAgentPerformanceStat
```

Then the name to use for the Dataset would be iAgentPerformanceStat

7.4.2. CCMS Summarized Historical Statistics

Historical Reporting applies the same data partitioning to each of the rolled up statistics views. The following table shows the data partitioning applied and only shows the interval version of these summarized historical statistics views; however if the report uses the Daily, Weekly or Monthly view name as the Dataset name, the same data partitioning will be applied:

View Name	Data Partition
iActivityCodeStat	Agent Application
iAgentByApplicationStat	Agent Application
iAgentBySkillsetStat	Agent Skillset
iAgentPerformanceStat	Agent
iApplicationStat	Application
iCDNStat	CDN
iDNISStat	DNIS
iRANMusicRouteStat	
iRouteStat	
iSkillsetStat	Application Skillset

7.4.3. CCMS Contact Statistics

The following table shows the data partitioning applied:

Name	Data Partition
eCSRCodeStat	Agent Application CDN Skillset
eCSRIntrinsicStat	
eCSRObserveBargeInStat	
eCSRStat	



7.4.4. CCMS Event Statistics

The following table shows the data partitioning applied:

Name	Data Partition
eAgentLoginStat	Agent
eCallByCallStatYYYYMMDD	

7.4.5. CCMS Configuration

The following table shows the data partitioning applied:

Name	Data Partition
AccessRights	
ActivityCode	
ActivityCodeByContactType	
ActivityCodeBySkillset	
Agent	Agent
Application	Application
ApplicationByScript	Application
ApplicationTemplate	
ApplicationThresholdTemplate	
CDN	CDN
CodeToMessageMap	
ContactType	Skillset
DNIS	DNIS
DNISThresholdTemplate	
Formula	
HistoricalStatCollection	Agent Application CDN DNIS Skillset
HistoricalStatDuration	
HistoricalStatStorage	
Provider	
Route	
RouteThresholdTemplate	
ScheduledSkillsetAssignment	Agent Skillset
ScheduledSupervisorAssignment	
Script	
ScriptVariableProperties	
ScriptVariables	
Site	

Name	Data Partition
Skillset	Skillset
SkillsetByAgent	Skillset
SkillsetByAssignment	Agent Skillset
SkillsetThresholdTemplate	
SummaryThresholdTemplate	
Supervisor	
SupervisorAgentAssignment	Agent
SupervisorByAssignment	
Treatment	
URIByAgent	
UserTemplate	
UserThresholdTemplate	
UserValidationReport	
UserValidationStatus	

7.4.6. CCMM Statistics

The following table shows the data partitioning applied:

Table / View Name	Data Partitioning Applied
CampaignAgentScriptResultsCallByCall	Skillset
CampaignCallByCallDetails	Skillset
CampaignSummary	Skillset
ContactsByContacts	Skillset
ContactsOutStanding	Skillset
ContactsOutstandingDetail	Skillset
ContactsOutstandingSumm	Skillset
ScriptSummary	

8. Appendix B: Writing the Report Query

To define a Dataset, a report query is required. The report query is an SQL Select statement. The following options are available:

- Type in the query or copy from a file.
- Use software to build the query and copy the text. If you have a version of Microsoft Excel, use the **Microsoft Query** wizard. See section Using Excel to Write the Dataset Query for details.
- Import the query from an existing report.

If you are comfortable writing SQL Select statements, the Report Builder offers an interface for entering and testing the query. The Report Builder also allows the query to be imported from an existing report.

Several other applications offer wizards to help create the query. It's beyond the scope of this document to offer details on multiple SQL builder software packages. Instead this document will focus on Microsoft Query included with Microsoft Excel.

Microsoft Excel includes Microsoft Query for adding external data to an Excel spreadsheet. The procedures for using Excel 2007 and Excel 2010 are included. Other versions of Excel will offer similar capabilities. If you are working with a different version of Excel, refer to the available documentation for that version.

8.1. SQL Select Statement

The SQL Select statement has mandatory and optional parts. The following shows part of the grammar covering just the items used in this document:

```
SELECT [TOP Number] * | Field [ AS ] Alias ]
[,Field [ AS Alias ]...]
FROM Table | View [ [ AS ] Alias ]
[WHERE Condition[,Condition ...]]
[GROUP BY Field [,Field ...]]
[ORDER BY Field [ASC | DESC][,Field [ASC | DESC] ...]]
```

When reading the grammar above:

- Items in square brackets are optional
- `Number` means a positive numeric value
- `*` means all fields

- The | character separates a list of items where only one can be used. For example ASC | DESC means select either ASC or DESC
- The ... means the same can be repeated multiple times
- Alias is a string used to reference the original Table or View
- ASC means ascending and DESC means descending

The minimal form of the SQL Select statement is:

```
SELECT * | Field [,Field ...]  
FROM Table | View
```

Knowing the fields, and the table or view, is enough to write a valid SQL Select statement. The problem is without filtering, the query will contain all of the records available. Use a WHERE clause or TOP Number to limit the number of records returned.

8.1.1. String delimiters

One of the differences between databases is how string values are delimited. For some, a string value can be enclosed in either double or single quotes. For Cache, a string value can only be enclosed in single quotes. The restriction occurs because Cache uses double quotes as the database element delimiter (i.e. Fields, tables, etc.).

If the string value contains a single quote, then use two single quotes to represent one single quote. For example, if the query contains a string value such as:

```
'Chris O'Connor'
```

Then the string value that works in the SQL Select statement will be written as:

```
'Chris O''Connor'
```

8.1.2. DateTime Fields

To enter a DateTime value, the value is entered as:

```
YYYY-MM-DD HH:mm:ss
```

The time is defined as 24 hour clock. For example, the date and time November 17th 2014 at 1:30 pm, would be:

```
2014-11-17 13:30:00
```



The ACCS databases use Timestamp fields, which are essentially DateTime fields. The SQL syntax provides a function for creating timestamp values. In all cases where a literal value for DateTime will be used, the value should be entered as follows:

```
{ts '2014-11-17 13:30:00'}
```

If the Time portion is not included, the Time value will be 00:00:00, meaning:

```
{ts '2014-11-17 00:00:00'} = {ts '2014-11-17'}
```

8.1.3. Reserved Terms

The CCMS performance statistics include the fields Timestamp and Time in the database views. In Cache these will cause an error because the terms Timestamp and Time are reserved terms.

To ensure that the SQL Select Statement will work, add the table or view name before the column name. For example, to select the Timestamp field from the view dAgentPerformanceStat:

```
SELECT dAgentPerformanceStat.Timestamp  
FROM dAgentPerformanceStat
```

If using an alias name instead of the view name, it is the alias name that is placed before the column name. For example, if the alias is iAgentPerformanceStat then the SQL would be:

```
SELECT iAgentPerformanceStat.Timestamp  
FROM dAgentPerformanceStat AS iAgentPerformanceStat
```

The use of AS before the alias is optional.

8.1.4. Using an Alias for a Table or View

CCMS provides Interval statistics and the rolled up statistics for Daily, Weekly and Monthly in different database views, using an alias name in place of the view name simplifies the editing required to reuse the SQL Select Statement.

The Microsoft Query builder will reuse the selected table or view as the alias name; however it does not need to match.

For example, if the query is:

```
SELECT dAgentPerformanceStat.Timestamp  
FROM dAgentPerformanceStat
```

The alias could be defined as iAgentPerformanceStat, and the updated query is:

```
SELECT iAgentPerformanceStat.Timestamp
```



```
FROM dAgentPerformanceStat AS iAgentPerformanceStat
```

The use of AS before the alias is optional.

Now to change the query to use the Monthly view, the update is limited to the view name and no other part of the SQL Select statement:

```
SELECT iAgentPerformanceStat.Timestamp  
FROM mAgentPerformanceStat AS iAgentPerformanceStat
```

8.1.5. Spaces in Column Names

If the database field names contain spaces then the field name must be delimited to ensure the SQL Select Statement can be correctly interpreted. For Cache, the delimiter for field names is double quotes.

For example, the CCMM database view ContactsOutstandingDetail defines field names using spaces. The SQL Select statement including By Day and Contact No would be written as:

```
SELECT ContactsOutstandingDetail."By Day",  
ContactsOutstandingDetail."Contact No"  
FROM vw.ContactsOutstandingDetail  
ContactsOutstandingDetail
```

8.1.6. Fields Shown in the Result

Some databases require that a field be specifically included in the SELECT for the SQL Select statement to be valid. For Cache, a field that is not listed can still be used.

For example, the following is valid for Cache databases:

```
SELECT iAgentPerformanceStat.AgentLogin  
FROM dAgentPerformanceStat AS iAgentPerformanceStat  
WHERE (iAgentPerformanceStat.Timestamp>={ts '2014-11-10  
00:00:00'})
```

The Timestamp field is not in the SELECT and will not be shown in the result; however it can still be used in the WHERE clause to filter.

8.1.7. Group By and Aggregates

To use GROUP BY in an SQL Select statement:

- Fields must be listed in the SELECT, cannot use *



- Field values must be aggregated
- To retain the original column names, alias names must be defined

When using a field name alias, the alias must be preceded by AS or the query will fail. The aggregates are functions, and the common aggregates are:

- SUM()
- MAX()
- MIN()

When the field value is not numeric, an aggregate must still be used. As long as the grouping still results in a single record value, using MIN or MAX will return the same value. If the grouping results in multiple values returned as a single record, MIN will return the smallest value and MAX the largest value. If the value is a name, this may not be the expected result, and the field may not be usable.

8.1.7.1.String Values are Uppercase

For Cache databases, an aggregate of a string field will result in the string becoming uppercase. To avoid this, another string function can be included in the query. The ACCS standard reports use the LTRIM () function, which removes whitespace characters from the left of the string.

For example, if the original query is:

```
SELECT iAgentPerformanceStat.Timestamp,  
iAgentPerformanceStat.AgentLogin,  
iAgentPerformanceStat.AgentSurName,  
iAgentPerformanceStat.AgentGivenName,  
iAgentPerformanceStat.LoggedInTime  
FROM dAgentPerformanceStat AS iAgentPerformanceStat
```

To add a GROUP BY for Timestamp and AgentLogin, the SQL Select Statement would be:

```
SELECT MIN(iAgentPerformanceStat.Timestamp) AS Timestamp,  
MIN(LTRIM(iAgentPerformanceStat.AgentLogin)) AS  
AgentLogin, MIN(LTRIM(iAgentPerformanceStat.AgentSurName))  
AS AgentSurName,  
MIN(LTRIM(iAgentPerformanceStat.AgentGivenName)) AS  
AgentGivenName, SUM(iAgentPerformanceStat.LoggedInTime) AS  
LoggedInTime  
FROM dAgentPerformanceStat AS iAgentPerformanceStat  
GROUP BY iAgentPerformanceStat.Timestamp,  
iAgentPerformanceStat.AgentLogin
```

8.2. Using Excel to Write the Dataset Query

If you are not comfortable writing SQL queries, Excel offers a query wizard that can be used to get the required SQL Select statement for the report Dataset.

8.2.1. Before you begin

The steps required to launch the Microsoft Query wizard may be different on different version of Excel. Please refer to documentation for the current version.

8.2.2. About this task

It is important to filter the data using either a TOP or WHERE clause. The Microsoft Query wizard does not offer filtering using TOP so the filtering must be defined as a WHERE clause with conditions. As well, it does not support using functions for the values (i.e. DateAdd(), Now(), etc.).

8.2.3. Procedure for Excel 2007 and 2010

1. On the **Data** tab, in the **Get External Data** group, click **From Other Sources**, and then click **From Microsoft Query**.
2. In the **Choose Data Source** dialog, click the **Databases** tab, select the ODBC data source to use from the list, check the option **Use the Query Wizard to create/edit queries** and click the **OK** button. The data sources listed are those configured in the **ODBC Data Source Administrator** under the **User DSN** and **System DSN** tabs
3. If requested, enter the **User Name** and **Password** for the data source and click the **OK** button.
4. In the **Query Wizard – Choose Columns** dialog, expand a table in the **Available tables and columns** list and double click each field you would like to move to the **Columns in your query** list. Optionally, the order of the items in the **Columns in your query** list can be adjusted by selecting an item and using the **Move up** or **Move down** buttons. Click the **Next** button.
5. In the **Query Wizard – Filter Data** dialog, optional set filters for the query. Click the **Next** button.
6. In the **Query Wizard – Sort Order** dialog, optional set the fields used for sorting the data. Click the **Next** button.
7. In the **Query Wizard – Finish dialog**, in the **What would you like to do next?** Group, select the **View data or edit query in Microsoft Query** radio button. Click the **Finish** button.
8. In the **Microsoft Query** dialog, click the **View** menu and select **SQL ...**
9. In the **SQL** dialog, right click the mouse in the **SQL Statement** text box and choose **Select All** from the popup menu, then right click the mouse in the **SQL Statement** text box and choose **Copy** from the popup menu.

The SQL Select statement is now stored in the clipboard and can be pasted into another application, such as the Dataset Query field in Microsoft Report Builder Builder or Notepad.

10. To optionally return the data to an Excel worksheet, Click the **File** menu and select **Return Data to Microsoft Excel**

8.2.4. Excel 2007 and 2010 Agent Performance Statistics Example

The following example writes an SQL Select statement for the Daily Agent Performance statistics view (dAgentPerformanceStat). The data will be filtered to the previous 7 days assuming the current date is 2014-11-17. The data will be sorted by Timestamp and then AgentLogin.

1. On the **Data** tab, in the **Get External Data** group, click **From Other Sources**, and then click **From Microsoft Query**.
2. In the **Choose Data Source** dialog, click the **Databases** tab, select **CCMS_PREVIEW_DSN** from the list, check the option **Use the Query Wizard to create/edit queries** and click the **OK** button.
3. If requested, enter the **User Name** and **Password** for the data source and click the **OK** button.
4. In the **Query Wizard – Choose Columns** dialog, expand the **dAgentPerformanceStat** table in the **Available tables and columns** list and double click each of the following fields to move them to the **Columns in your query** list:

- Timestamp
- AgentLogin
- AgentGivenName
- AgentSurName
- CallsOffered
- CallsAnswered
- LoggedInTime
- TalkTime
- HoldTime
- PostCallProcessingTime
- NotReadyTime
- WaitingTime

Click the **Next** button.

5. In the **Query Wizard – Filter Data** dialog:
 - a. In the **Columns to filter** list select **Timestamp** and in the **Only include rows where** group select is **greater than and equal to** as the condition and type a date like 2014-11-01 has the value
 - b. In the **Only include rows where** group select **And**, then is **less than and equal to** as the condition and type a date like 2014-11-07 has the value
 - c. Click the **Next** button.
6. In the **Query Wizard – Sort Order** dialog:
 - a. Under **Sort by** select **Timestamp** from the list
 - b. Under **Then by** select **AgentLogin** from the list
 - c. Click the **Next** button.

7. In the **Query Wizard – Finish** dialog, in the **What would you like to do next?** Group, select the **View data or edit query in Microsoft Query** radio button. Click the **Finish** button.
8. In the **Microsoft Query** dialog, click the **View** menu and select **SQL ...**

The SQL Select statement generated is:

```
SELECT dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin,
dAgentPerformanceStat.AgentSurName,
dAgentPerformanceStat.AgentGivenName,
dAgentPerformanceStat.CallsOffered,
dAgentPerformanceStat.CallsAnswered,
dAgentPerformanceStat.LoggedInTime,
dAgentPerformanceStat.TalkTime,
dAgentPerformanceStat.HoldTime,
dAgentPerformanceStat.PostCallProcessingTime,
dAgentPerformanceStat.NotReadyTime,
dAgentPerformanceStat.WaitingTime
FROM dbo.dAgentPerformanceStat dAgentPerformanceStat
WHERE (dAgentPerformanceStat.Timestamp>={ts '2014-11-10
00:00:00'} And dAgentPerformanceStat.Timestamp<={ts
'2014-11-16 00:00:00'})
ORDER BY dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin
```

9. In the **SQL** dialog, right click the mouse in the **SQL Statement** text box and choose **Select All** from the popup menu, then right click the mouse in the **SQL Statement** text box and choose **Copy** from the popup menu.

The SQL Select statement is now stored in the clipboard and can be pasted into another application, such as the Dataset Query field in Microsoft Report Builder or Notepad.

10. To optionally return the data to an Excel worksheet, Click the **File** menu and select **Return Data to Microsoft Excel**

8.2.5. Excel 2007 and 2010 Activity Code Statistics Example

The following example writes an SQL Select statement against the Daily Activity Code statistics view (dActionCodeStat). The data will be filtered to the previous 7 days assuming the current date is 2014-11-17 and to remove the 'System_Application'. The data will be sorted by Timestamp, then AgentLogin and then ActivityCode.

The 'System_Application' is used when the Activity is a Not Ready Reason Code, Activities entered while active on a contact are stored against the application that routed the contact to the Agent.

1. On the **Data** tab, in the **Get External Data** group, click **From Other Sources**, and then click **From Microsoft Query**.
2. In the **Choose Data Source** dialog, click the **Databases** tab, select **CCMS_PREVIEW_DSN** from the list, check the option **Use the Query Wizard to create/edit queries** and click the **OK** button.
3. If requested, enter the **User Name** and **Password** for the data source and click the **OK** button.
4. In the **Query Wizard – Choose Columns** dialog, expand the **dActivityCodeStat** table in the **Available tables and columns** list and double click each of the following fields to move them to the **Columns in your query list**:
 - Timestamp
 - ActivityCode
 - ActivityCodeName
 - AgentLogin
 - AgentGivenName
 - AgentSurName
 - ActivityTime
 - OccurrencesClick the **Next** button.
5. In the **Query Wizard – Filter Data** dialog:
 - a. In the **Columns to filter** list select **Timestamp** and in the **Only include rows where** group select is **greater than and equal to** as the condition and type a date like 2014-11-10 has the value
 - b. In the **Only include rows where** group select **And**, then is **less than and equal to** as the condition and type a date like 2014-11-16 has the value
 - c. In the **Columns to filter** list select **Application** and in the **Only include rows where** group select **does not equal** as the condition and select **System_Application** from the list has the value
 - d. Click the **Next** button.
6. In the **Query Wizard – Sort Order** dialog:
 - a. Under **Sort by** select **Timestamp** from the list
 - b. Under **Then by** select **AgentLogin** from the list
 - c. Under **Then by** select **ActivityCode** from the list
 - d. Click the **Next** button.
7. In the **Query Wizard – Finish** dialog, in the **What would you like to do next?** group, select the **View data or edit query in Microsoft Query** radio button. Click the **Finish** button.
8. In the **Microsoft Query** dialog, click the **View** menu and select **SQL ...**

The SQL Select statement generated is:

```
SELECT dActivityCodeStat.Timestamp,  
dActivityCodeStat.ActivityCode,  
dActivityCodeStat.ActivityCodeName,  
dActivityCodeStat.AgentLogin,  
dActivityCodeStat.AgentGivenName,  
dActivityCodeStat.AgentSurName,  
dActivityCodeStat.ActivityTime,  
dActivityCodeStat.Occurrences  
FROM dbo.dActivityCodeStat dActivityCodeStat  
WHERE (dActivityCodeStat.Timestamp>={ts '2014-11-10  
00:00:00'} AND dActivityCodeStat.Timestamp<={ts '2014-11-  
16 00:00:00'}) AND  
(dActivityCodeStat.Application<>'System_Application')  
ORDER BY dActivityCodeStat.Timestamp,  
dActivityCodeStat.AgentLogin,  
dActivityCodeStat.ActivityCode
```

9. In the **SQL** dialog, right click the mouse in the **SQL Statement** text box and choose **Select All** from the popup menu, then right click the mouse in the **SQL Statement** text box and choose **Copy** from the popup menu.

The SQL Select statement is now stored in the clipboard and can be pasted into another application, such as the Dataset Query field in Microsoft Report Builder Builder or Notepad.

10. To optionally return the data to an Excel worksheet, Click the **File** menu and select **Return Data to Microsoft Excel**

8.2.6. Modifying the Query in Microsoft Query

It is possible in some cases to modify the query in the Microsoft Query; however the graphical representation of the query will be disabled. Once the graphical representation becomes disable, all changes must be done by clicking click the **View** menu and selecting **SQL ...** from the available items.

This only limits the features of Microsoft Query, and it retains the ability to:

- Test the query
- See the results
- Return the results to an Excel spreadsheet

8.2.7. Modifying the Query Filtering

A query generated with specific Date values will work; however some of the filtering can be improved to ensure editing the report in the future will still return data. To satisfy the Microsoft Query wizard, the values to filter the data to the previous 7 days were entered as literal date value.

8.2.7.1. Modify Literal Dates to be Relative Dates

Looking at the Agent Performance Statistics SQL Select statement, the current WHERE clause is:

```
WHERE (iAgentPerformanceStat.Timestamp>={ts '2014-11-10
00:00:00'} And iAgentPerformanceStat.Timestamp<={ts '2014-
11-16 00:00:00'})
```

The values can be replaced using the DateAdd() function of SQL. To get the previous 7 days from the time the SQL Select Statement is sent to the database, replace the values as follows:

```
{ts '2014-11-10 00:00:00'} => DateAdd('d', -8, Now())
{ts '2014-11-16 00:00:00'} => DateAdd('d', -1, Now())
```

The updated WHERE clause is:

```
WHERE (dAgentPerformanceStat.Timestamp>= DateAdd('d', -8,
Now()) And dAgentPerformanceStat.Timestamp<= DateAdd('d',
-1, Now()))
```

8.2.7.2. Adding TOP to the Query

To prevent the returned data exceeding a set number of records, add TOP to the SQL Select Statement. The format is TOP Number where the number is a positive numeric value. If the maximum number of records to return is 100, then the text TOP 100 would be placed after SELECT in the query.

The records returned are the first set of records based on the sort order. By sorting the data correctly it is possible to use this to get the first number of records with the lowest (ASC) or greatest (DESC) value.

The actual number of records may need to be changed to get a greater volume of data for testing, or to get details required to build the report. The general goal is to ensure that:

- Data is returned quickly while designing the report
- The number of records are limited to decrease the load on the database

Before the report is imported to ACCS, this value may need to be changed or removed. This is strictly a design choice, and as seen here, TOP could be used to set a maximum limit for the number of records to include on the report.

8.2.7.3. Updated Agent Performance Statistics Example

After applying these updates, the SQL Select Statement will be:

```
SELECT TOP 100
dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin,
dAgentPerformanceStat.AgentSurName,
dAgentPerformanceStat.AgentGivenName,
dAgentPerformanceStat.CallsOffered,
dAgentPerformanceStat.CallsAnswered,
dAgentPerformanceStat.LoggedInTime,
dAgentPerformanceStat.TalkTime,
dAgentPerformanceStat.HoldTime,
dAgentPerformanceStat.PostCallProcessingTime,
dAgentPerformanceStat.NotReadyTime,
dAgentPerformanceStat.WaitingTime
FROM dbo.dAgentPerformanceStat dAgentPerformanceStat
WHERE (dAgentPerformanceStat.Timestamp>= DateAdd('d', -8,
Now()) And dAgentPerformanceStat.Timestamp<= DateAdd('d',
-1, Now()))
ORDER BY dAgentPerformanceStat.Timestamp,
dAgentPerformanceStat.AgentLogin
```

8.2.7.4. Updated Activity Code Statistics Example

After applying these updates, the SQL Select Statement will be:

```
SELECT TOP 100
dActivityCodeStat.Timestamp,
dActivityCodeStat.ActivityCode,
dActivityCodeStat.ActivityCodeName,
dActivityCodeStat.AgentLogin,
dActivityCodeStat.AgentGivenName,
dActivityCodeStat.AgentSurName,
dActivityCodeStat.ActivityTime,
dActivityCodeStat.Occurrences
FROM dbo.dActivityCodeStat dActivityCodeStat
WHERE (dActivityCodeStat.Timestamp>= DateAdd('d', -8,
Now()) And dActivityCodeStat.Timestamp<= DateAdd('d', -1,
Now())) AND
(dActivityCodeStat.Application<>'System_Application')
ORDER BY dActivityCodeStat.Timestamp,
dActivityCodeStat.AgentLogin,
dActivityCodeStat.ActivityCode
```

8.2.8. Adding Group By to the Query

The CCMS views define primary keys that make up a unique record.

8.2.8.1. Updated Agent Performance Statistics Example

Using the table of Primary Keys for CCMS Summarized Historical Statistics, the primary keys for any of the AgentPerformanceStat views are:

- Timestamp
- UserID/AgentLoginID
- SupervisorUserID/SupervisorLoginID

The example does not show the Supervisor details; however will show the Timestamp and Agent details. The Agent can have multiple Supervisors if the agent is reassigned to different supervisors.

To generate a query that will remove the Supervisor but retain the data, the SQL Select statement will be updated to:

- Group by the Timestamp and AgentLogin fields
- Aggregate the values

After applying these updates, the SQL Select Statement will be:

```
SELECT TOP 100
MIN(dAgentPerformanceStat.Timestamp) AS Timestamp,
MIN(LTRIM(dAgentPerformanceStat.AgentLogin)) AS
AgentLogin,
MIN(LTRIM(dAgentPerformanceStat.AgentSurName)) AS
AgentSurName,
MIN(LTRIM(dAgentPerformanceStat.AgentGivenName)) AS
AgentGivenName,
SUM(dAgentPerformanceStat.CallsOffered) AS CallsOffered,
SUM(dAgentPerformanceStat.CallsAnswered) AS CallsAnswered,
SUM(dAgentPerformanceStat.LoggedInTime) AS LoggedInTime,
SUM(dAgentPerformanceStat.TalkTime) AS TalkTime,
SUM(dAgentPerformanceStat.HoldTime) AS HoldTime,
SUM(dAgentPerformanceStat.PostCallProcessingTime) AS
PostCallProcessingTime,
SUM(dAgentPerformanceStat.NotReadyTime) AS NotReadyTime,
SUM(dAgentPerformanceStat.WaitingTime) AS WaitingTime
FROM dbo.dAgentPerformanceStat dAgentPerformanceStat
WHERE (dAgentPerformanceStat.Timestamp>= DateAdd('d', -8,
Now())) And dAgentPerformanceStat.Timestamp<= DateAdd('d',
-1, Now()))
```



```
GROUP BY dAgentPerformanceStat.Timestamp,  
dAgentPerformanceStat.AgentLogin  
ORDER BY dAgentPerformanceStat.Timestamp,  
dAgentPerformanceStat.AgentLogin
```

8.2.8.2. Updated Activity Code Statistics Example

Using the table of Primary Keys for CCMS Summarized Historical Statistics, the primary keys for any of the ActivityCodeStat views are:

- Timestamp
- UserID/AgentLoginID
- ApplicationID
- ActivityCode

The example does not show the Application details; however will show the Timestamp, Agent and Activity Code details. The Activity code can be used by multiple Agents on contacts routed by multiple Applications for each Timestamp.

To generate a query that will remove the Supervisor but retain the data, the SQL Select statement will be updated to:

- Group by the Timestamp, ActivityCode and AgentLogin fields
- Aggregate the values

After applying these updates, the SQL Select Statement will be:

```
SELECT TOP 100  
MIN(dActivityCodeStat.Timestamp) AS Timestamp,  
MIN(LTRIM(dActivityCodeStat.ActivityCode)) AS  
ActivityCode,  
MIN(LTRIM(dActivityCodeStat.ActivityCodeName)) AS  
ActivityCodeName,  
MIN(LTRIM(dActivityCodeStat.AgentLogin)) AS AgentLogin,  
MIN(LTRIM(dActivityCodeStat.AgentGivenName)) AS  
AgentGivenName,  
MIN(LTRIM(dActivityCodeStat.AgentSurName)) AS  
AgentSurName,  
SUM(dActivityCodeStat.ActivityTime) AS ActivityTime,  
SUM(dActivityCodeStat.Occurrences) AS Occurrences  
FROM dbo.dActivityCodeStat dActivityCodeStat  
WHERE (dActivityCodeStat.Timestamp>= DateAdd('d', -8,  
Now())) And dActivityCodeStat.Timestamp<= DateAdd('d', -1,  
Now())) AND  
(dActivityCodeStat.Application<>'System_Application')
```



```
GROUP BY dActivityCodeStat.Timestamp,  
dActivityCodeStat.ActivityCode,  
dActivityCodeStat.AgentLogin  
ORDER BY dActivityCodeStat.Timestamp,  
dActivityCodeStat.AgentLogin,  
dActivityCodeStat.ActivityCode
```

8.3. Import Query from Existing Report

The ACCS standard reports are defined with a design time report query. If the query is imported from an existing ACCS standard report, it is important to remember this is not the runtime query for the report.

8.3.1. Procedure

1. Click the **Query Designer** button
2. In the **Query Designer** dialog, click the **Import ...** menu item
3. In the **Import Query** dialog, navigate to an existing report file, select the file and click the **Open** button
4. In the **Import Query** dialog select a Dataset, confirm the query is correct and click the **Import** button.

9. Appendix C: Page Setup

9.1. Page Size

The Page Size set for a report represents the size and orientation used when printing or exporting the report. The following table shows the paper sizes for A4 and Letter in different orientations:

Paper	Centimeters (cm)		Inches (in)	
	Width	Height	Width	Height
A4 – Portrait	21.0	29.7	8.26	11.69
A4 – Landscape	29.7	21.0	11.69	8.26
Letter – Portrait	21.5	27.9	8.5	11.0
Letter - Landscape	27.9	21.5	11.0	8.5

ACCS standard reports need to fit both A4 and Letter. To achieve this, the size is reduced to the smallest measurements of both:

Paper	Centimeters (cm)		Inches (in)	
	Width	Height	Width	Height
Portrait	21.0	27.9	8.26	11.0
Landscape	27.9	21.0	11.0	8.26

9.2. Margins

Printers can have a non-printable region, and the size of this region differs from printer to printer. The standard Historical Reports allow for a 0.55 in or 1.4 cm margin on all sides.

9.2.1. Margins and the Report Viewer

The Report Viewer does not show report margins. If the printer margins are placed into the report margins only, objects placed at the far edge of the page will be shown at the very edge of the viewer, making it difficult to read the details.

For this reason, it is recommended to place object at least 0.3 cm from the edge of the visible page, which will be referred as the interactive margin. The remaining printer margin value is then entered as the report margins.

9.2.2. Recommended Settings

The recommendation is to use the following values, where the Interactive Margin is the position of report items from the edge of the page and Effective Margin is the amount of space reserved for print margins:

	Centimeters (cm)			Inches (in)		
	Report Margin	Interactive Margin	Print Margin	Report Margin	Interactive Margin	Print Margin
Left	1.1	0.3	1.4	0.43	0.12	0.55
Right	1.1	0.3	1.4	0.43	0.12	0.55
Top	1.1	0.3	1.4	0.43	0.12	0.55
Bottom	1.1	0.3	1.4	0.43	0.12	0.55

9.2.3. Moving and Resizing Items

The arrow keys can be used to adjust the position and size of selected items on the design surface of the Report Builder:

Key Combinations	Change to selected report item
Arrow keys	Move 1/8 in or 0.125 in (0.3175 cm).
CTRL+Arrow keys	Nudge 1/80 in or 0.0125 (0.03175 cm).
SHIFT+Arrow keys	Increase or decrease the size by 1/8 in.
CTRL+SHIFT+Arrow keys	Increase or decrease the size by 1/80 in.

9.3. Interactive Size

The interactive size represents the size shown to the user when viewed in the report viewer.

The interactive size determines the maximum size of the report body per page. The interactive size is reduced by the size of:

- Margins
- Header Height
- Footer Height

9.3.1. Fit to Page

SSRS reporting does not provide a fit to page feature, so having an interactive size larger than the page size will result in additional pages when the report is printed or exported. Since the ACCS standard reports must work on unknown printers, the standard reports attempt to use settings that will fit the report to a single page.



When building custom reports, the target printers may be known. For printing reports, depending on the printer, it may be possible to request the printer to scale the report to fit the paper page size. Refer to the printer documentation for details.

For exported reports, it is possible to use the target applications to fit the report to the printer page size. See the applications documentation for details.

9.3.2. Recommend Settings

It is not possible to provide a single set of recommended values for the Interactive Size since it depends on multiple items. The following calculations can be used as the recommended Interactive Size Height and Width:

Interactive	Calculation
Width	Page Size Width - (Margins Left + Margins Right)
Height	Page Size Height - (Margins Top + Margins Bottom + Header Height + Footer Height)

For example, a report includes a Header and Footer with the settings as follows:

Page Size:

Width = 21.5 cm

Height = 27.9 cm

Margins:

Left = 1.1 cm

Right = 1.1 cm

Top = 1.1 cm

Bottom = 1.1 cm

Header Height = 2.5 cm

Footer Height = 2 cm

The recommended Interactive Size settings would be:

Width = $21.5 - (1.1 + 1.1) = 21.5 - 2.2 = \mathbf{19.3 \text{ cm}}$

Height = $27.9 - (1.1 + 1.1 + 2.5 + 2) = 27.9 - 6.7 = \mathbf{21.2 \text{ cm}}$

10. Appendix D: Document Map

A document map provides a set of navigational links to report items. A user can click links in the document map to jump to the report page that displays that item.

How the document map appears is based on the target:

Target	Document Map
Report Viewer	Navigation Tree
PDF	Bookmarks
Excel	Worksheet named Document Map with Hyperlinks
Word	Table of contents with Hyperlinks
Print	Does not print

10.1. Creating a Document Map

A document map entry can be associated with:

- Report items
- Report groups

The steps required to add a document map are different for a report item and a report group.

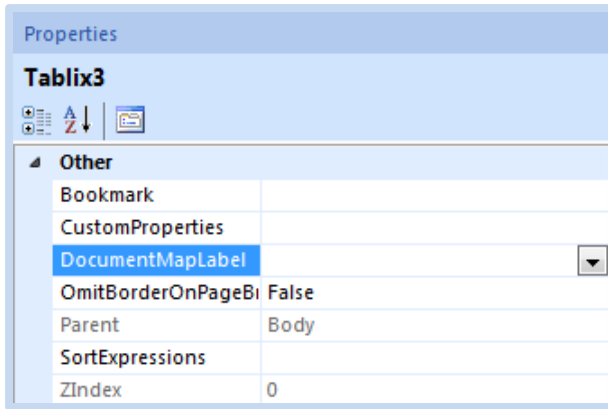
10.1.1. Report Item

10.1.1.1. Before you begin

Make sure that the **Properties** view is visible. If **Properties** is not visible select the **View** menu item and check the box for **Properties**.

10.1.1.2. Procedure

1. Select the report item.
2. In the **Properties** view, select the **Categorized** icon and locate the **Other** category
3. For the value beside **DocumentMapLabel**
 - a. Static Text
 - i. Type text to display, value will not change
 - b. Generated Text
 - i. Click the **arrow** beside the value field
 - ii. Select **<Expression ...>** from the list
 - iii. Enter an expression that will return the text to use in the **Set expression for: DocumentMapLabel** field
 - iv. Click **OK** to accept the changes and close the **Expression** dialog



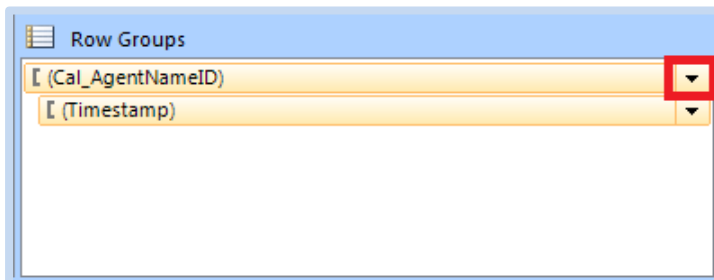
10.1.2. Report Group Procedure

10.1.2.1. Before you begin

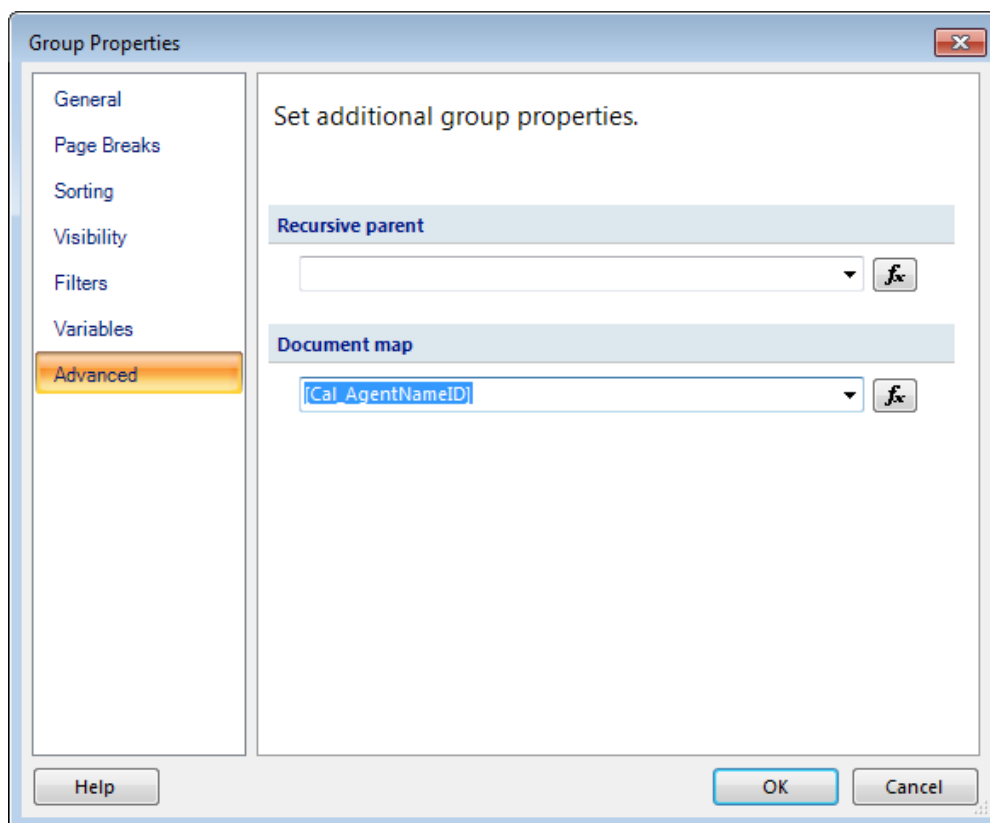
Make sure the **Grouping** view is visible. If **Grouping** is not visible select the **View** menu item and check the box for **Grouping**.

10.1.2.2. Procedure

1. Click the arrow beside the group in the **Row Groups** list and select **Group Properties** from the popup menu.



2. Select **Advanced** in the **Group Properties** dialog.
3. For the value under **Document map**
 - a. Dataset field
 - i. Click the arrow beside the value field
 - ii. Select a Dataset field from the list
 - b. Generated text
 - i. Click the **fx** button beside the value field
 - ii. Enter an expression that will return the text to use in the **Set expression for: DocumentMapLabel** field
 - iii. Click **OK** to accept the changes and close the **Expression** dialog



4. Click **OK** to accept the changes and close the **Group Properties** dialog

10.2. Limitations

As the number of items in the document increases, the time to generate the report and navigate pages increases. Having a large number of items in a document map can fail when exported to Excel and Word. To avoid these limitations, the ACCS standard reports were design to:

- Limit the number of items in the document map to 5000
- Remove the document map when exporting to Excel and Word.

10.2.1. Limit number of items

To enforce a document map limit:

- Determine if the limit is exceeded using a report variable
- Using expression to conditional remove the document map if limit is exceeded

10.2.1.1. Procedure

This procedure assumes the document map is only for a report group against the calculated field Cal_AgentNameID from the Dataset named dAgentPerformanceStat.

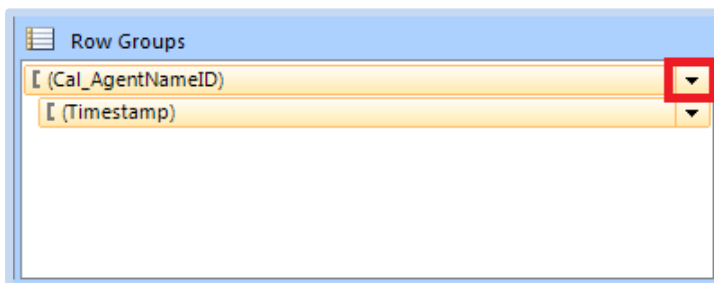
1. Right click the mouse off the visible report page (i.e. Above, Below, Left or Right) and select **Report Properties ...** from the popup menu
2. Select Variables in the **Report Properties** dialog
3. Click the **Add** button and enter:

- a. **Name:** DocumentMapExceedsLimit
- b. Click the **fx** button beside the value field
- c. Select the text in the **Set expression for: Value** field and press the **Delete** key to remove all text.
- d. Paste the following expression text into the **Set expression for: Value** field:

```
=(CountDistinct (Fields!Cal_AgentNameID.Value,
"dAgentPerformanceStat") > 1000)
```

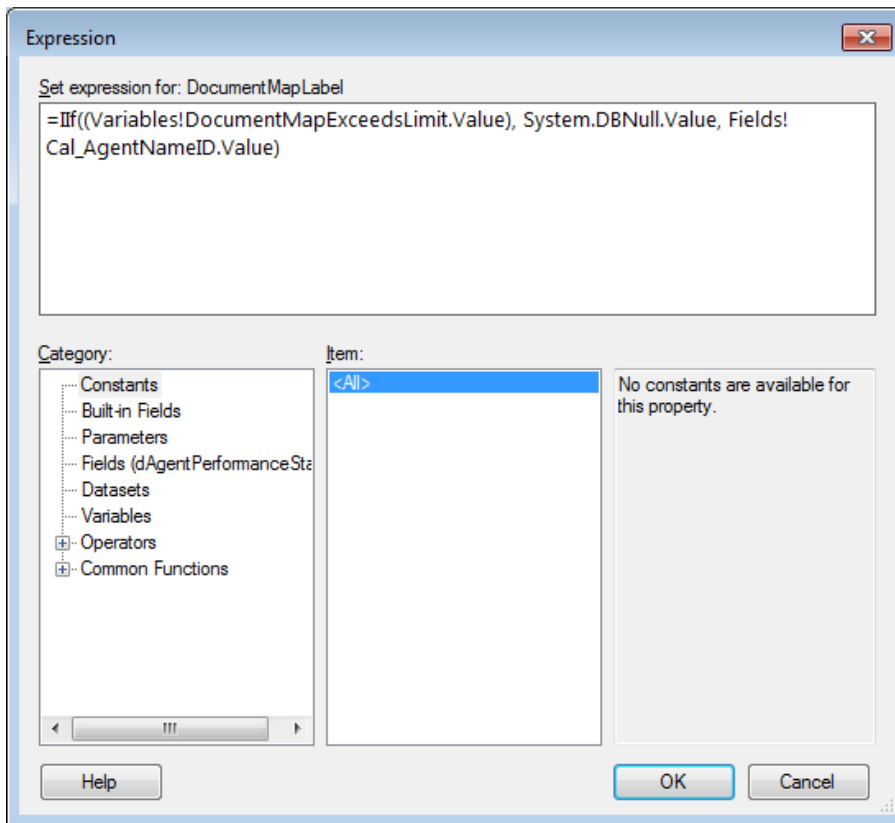
Where the limit is set to 1000 items

- e. Click **OK** to accept the changes and close the **Expression** dialog
4. Click **OK** to accept the changes and close the **Text Box Properties** dialog
5. Click the arrow beside the group in the **Row Groups** list and select **Group Properties** from the popup menu.



6. Select **Advanced** in the **Group Properties** dialog.
7. For the value under **Document map**
 - a. Click the **fx** button beside the value field
 - b. Select the text in the **Set expression for: DocumentMapLabel** field and press the **Delete** key to remove all text.
 - c. Paste the following expression text into the **Set expression for: DocumentMapLabel** field:


```
=IIf ( (Variables!DocumentMapExceedsLimit.Value) ,
System.DBNull.Value, Fields!Cal_AgentNameID.Value)
```
 - d. Click **OK** to accept the changes and close the **Expression** dialog



8. Click **OK** to accept the changes and close the **Group Properties** dialog

10.2.2. Remove when exporting to Excel and Word

The expression to conditional remove the document map is:

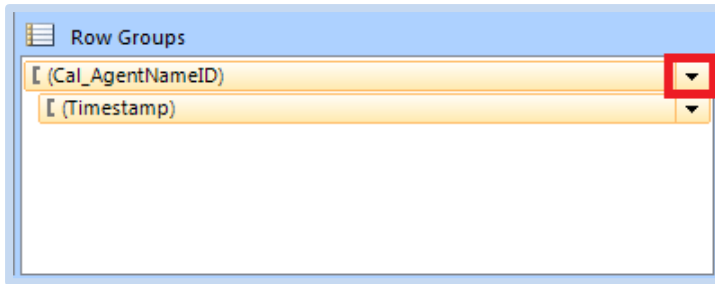
```
=IIf((Globals!RenderFormat.Name ="WORDOPENXML" OR
Globals!RenderFormat.Name ="EXCELOPENXML"),
System.DBNull.Value, <Document Map Value>)
```

Where <Document Map Value> is the value used when the report is not rendered to Excel or Word.

10.2.2.1. Procedure

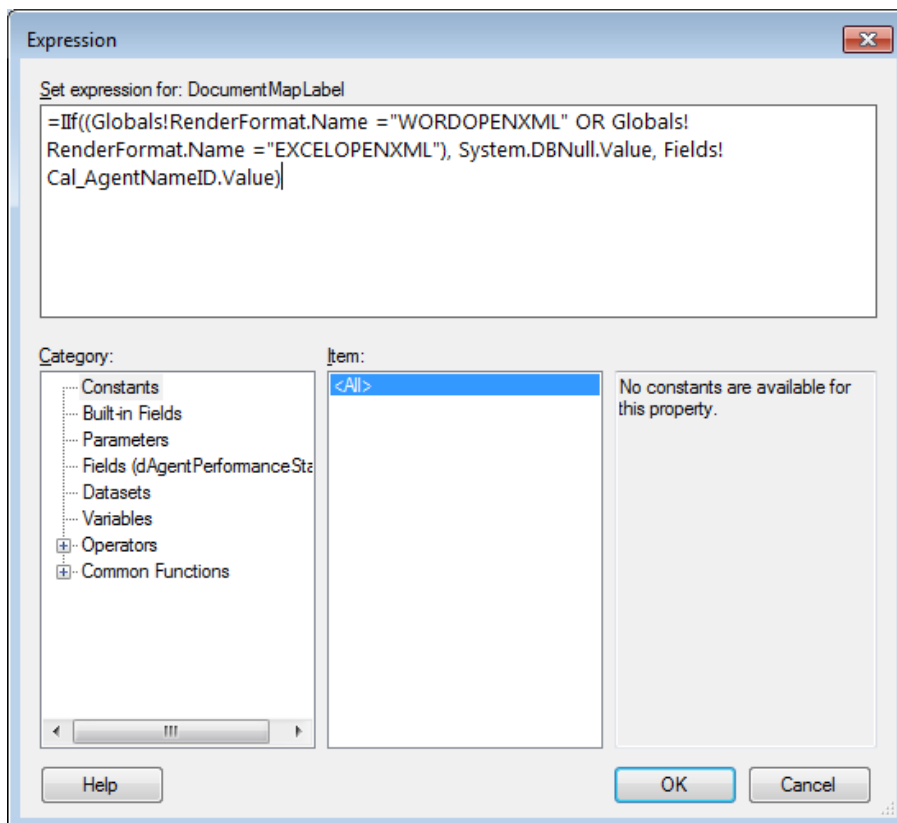
This procedure assumes the document map is for a report group against the calculated field Cal_AgentNameID. For different fields, replace this field name in the expression. The same expression can be applied to report item document maps.

1. Click the arrow beside the group in the **Row Groups** list and select **Group Properties** from the popup menu.



2. Select **Advanced** in the **Group Properties** dialog.
3. For the value under **Document map**
 - a. Click the **fx** button beside the value field
 - b. Select the text in the **Set expression for: DocumentMapLabel** field and press the **Delete** key to remove all text.
 - c. Paste the following expression text into the **Set expression for: DocumentMapLabel** field:


```
=IIf( (Globals!RenderFormat.Name ="WORDOPENXML" OR
          Globals!RenderFormat.Name ="EXCELOPENXML"),
          System.DBNull.Value, Fields!Cal_AgentNameID.Value)
```
 - d. Click **OK** to accept the changes and close the **Expression** dialog



4. Click **OK** to accept the changes and close the **Group Properties** dialog

11. Appendix E: ACCS Integration

ACCS Historical Reporting is designed to use and update specific items on a report template at runtime. These items must have a specific name or follow a defined naming convention.

11.1. Runtime Variables

Runtime variables have their values replaced at runtime with values defined in Historical Reporting. Variables are not available when writing expressions for calculated fields for a Dataset.

11.1.1. Naming Convention

Runtime variables start with “report_”, with the exception of “site_name_id”, and for this reason user created variables should avoid names starting with “report_”. The list of runtime variables can increase in each release and all future runtime variables will start with “report_”.

11.1.2. General

Name	Description
report_title	The report title defined in Historical reporting
report_user	The name of the user running the report
report_language	The language used for localization
report_filepath	The file path of the report template

11.1.3. Server

Name	Description
site_id_name	The CCMS server display name defined in configuration
report_server_majorversion	The CCMS server major version
report_server_minorversion	The CCMS server minor version
report_server_servicepack	The CCMS server service pack level
report_server_type	The CCMS server type
report_server_subtype	The CCMS server subtype
report_server_associated_cct	The CCMS server has an associated CCT server
report_server_associated_cm	The CCMS server has an associated CM server
report_feature_heteronetworking	The CCMS server supports HeteroNetworking
report_feature_networking	The CCMS server supports Networking
report_feature_openqueue	The CCMS server supports OpenQueue
report_feature_multiplicity	The CCMS server supports Multiplicity
report_feature_agentgreeting	The CCMS server supports Agent Greeting
report_feature_offsiteagent	The CCMS server supports Offsite Agent
report_feature_impresence	The CCMS server supports IMPresence
report_feature_voiceproxy	The CCMS server supports Voice Proxy
report_feature_ssoenabled	The CCMS server has SSO Enabled
report_feature_amszoning	The CCMS server supports AMS Zoning

11.1.4. Internationalisation

Name	Description
report_fontname	The name of the font based on the client's settings
report_fontname_numeric	The name of the numeric font based on the client's settings.
report_format_firstlastnameorder	The display order of names based on user preference. The valid values are: <ul style="list-style-type: none"> • 0 = First Last • 1= Last, First
report_format_nameidorder	The display order of name and ID based on user preference. The valid values are: <ul style="list-style-type: none"> • 0 = Name – ID • 1= ID – Name

11.2. Runtime Parameters

Runtime parameters have their values replaced at runtime by values defined in ACCS Historical Reporting. Parameters are available when writing expressions for calculated fields for a Dataset.

11.2.1. Naming Convention

Runtime parameters start with “report_”, and for this reason user created parameters should avoid names starting with “report_”. The list of runtime parameters can increase in each release and all future runtime parameters will start with “report_”.

Name	Description
report_datarange	The numeric value related to the data range selected in the Historical Reporting application. The valid values are: <ul style="list-style-type: none"> • 0 =Not applicable • 1 =Interval • 2 = Daily • 4 = Weekly • 8 = Monthly
report_interval	An internationalized string representing the collection period for the report.
report_interval_end	DateTime object representing the collection period end date and time for report.
report_interval_start	DateTime object representing the collection period start date and time for report.
report_timeoffset	The amount of time, in minutes, added to the current data to convert it from server timezone to the client’s timezone. This value only applies to Interval and Event based statistics.
report_interval_rangetype	The numeric value related to the interval range type. The valid

Name	Description
	values are: <ul style="list-style-type: none"> • -1 = Between • 0 = All Interval s • 1 = Previous Intervals

11.2.1.1. *Parameter Properties*

Use the following properties when adding a runtime parameter, where <Name> is the name of the runtime parameter and <Valid Value> is a value used while designing the report and fit the valid values for the runtime parameter:

General

Parameter name: <Name>

Data type: Text

Allow null value: Checked

Select parameter visibility: Internal

Default Values

Select from one of the following options: Specify values

Add value: <Valid Value>

11.3. Runtime Query Parameters

A runtime query is built from a set of parameters that have their values read at runtime. These parameters do not have their values replaced at runtime.

Once the original Dataset query has been used to create a set of runtime query parameters, the details of these parameters must be maintained when the original Dataset query is updated. Since the Report Builder continues to use the original Dataset query, problems with these parameters will not be seen until the report is run within ACCS.

The following table shows when runtime query parameters must be defined and which parameters are required:

Feature	Timestamp Parameter	Row Filter Parameter	Query Parameter
Data Partitioning	No	Yes	Yes
Data Range	No *	Yes	Yes
Data Range Applies to Multiple Fields	Yes	Yes	Yes

* This parameter is required for ACCS 6.4 SP13.

11.3.1. Naming Convention

Runtime query parameters start with “report_”, and for this reason user created parameters should avoid names starting with “report_”. The list of runtime query parameters can increase in each release and all future runtime query parameters will start with “report_”.

Name	Description
report_timestamp_<Dataset Name>	Provides the timestamp field, or a comma separated list of timestamp fields, used to apply the Data Range filtering to the runtime query.
report_rowfilter_<Dataset Name>[Name]	Provides the details of which database fields should be filtered.
Report_query_<Dataset Name>	Provides the query used to build the runtime query.

Where <Dataset Name> is the name of the Dataset for which the parameter applies, and [Name] is an optional name to make the parameter unique allowing for multiple parameters to be associated with the same Dataset.

11.3.2. Timestamp Parameter

Provides the timestamp field, or a comma separated list of timestamp fields, used to apply the Data Range filtering to the runtime query.

11.3.2.1. Rules applied to determine runtime value

The following rules are used to determine the runtime value. The rules are checked in order, and the first rule to match will set the value:

Rule	Value used
Available Value label matches Data Range	Matching label value
Available Value label does not match Data Range	First Available Value that is not Null or Empty
ACCS SP14 or later:	Timestamp field defined during import

Rule	Value used
Available Value not found	
Default	Data Range is not applied to the runtime query.

11.3.2.2. *Parameter Properties*

Use the following properties for the parameter, where <Dataset Name> is the name of the Dataset for which the parameter applies:

General

Parameter name: report_timestamp_<Dataset Name>

Data type: Text

Allow null value: Checked

Select parameter visibility: Internal

Available Values

Select from one of the following options: Specify values

Add value:

Label	Value
None	(Null)

Add at least one of the following:

Label	Value
Interval	Field to use when report interval is Interval
Daily	Field to use when report interval is Daily
Weekly	Field to use when report interval is Weekly
Monthly	Field to use when report interval is Monthly
Any valid label	Field to use if the labels Interval, Daily, Weekly or Monthly are not found.

Default Values

Select from one of the following options: Specify values

Add value: (Null)

11.3.3. Row Filter Parameter

The row filter parameter provides a reference in the query where filtering is applied, and the details of which database fields should be filtered. The parameter can define a WHERE clause that is always applied to the runtime query.

When adding a query to the query parameter, replace the existing WHERE clause with the name of the row filter parameter, formatted as:

{?report_rowfilter_<Dataset Name>}

11.3.3.1. Parameter Properties

Use the following properties for the parameter, where <Dataset Name> is the name of the Dataset for which the parameter applies:

General

Parameter name: report_rowfilter_<Dataset Name>

Data type: Text

Allow null value: Checked

Select parameter visibility: Internal

Available Values

Select from one of the following options: Specify values

Add values:

Label	Value
TableAlias	* TableAlias of filters to apply, can be a comma separated list
FieldName	* FieldName of filters to apply, can be a comma separated list
Default	Default Or A default filter that is always included. The value must start with WHERE; otherwise the value is ignored. This WHERE clause is always joined to other filters using AND.



Where * means all Table Aliases or Field Names

Default Values

Select from one of the following options: Specify values

Add value: Default or select one of the Available Values

11.3.3.2. Example Settings for Available Values

These examples will be based on the dActivityCodeStat view. This view can have the Data Range connected to the database field Timestamp.

1.1.1.1.1.1 Apply Data Range and Data Partitioning

Label	Value
TableAlias	*
FieldName	*

1.1.1.1.1.2 Apply Data Range filtering only

Label	Value
TableAlias	*
FieldName	Timestamp

11.3.4. Query Parameter

Provides the query used to build the runtime query. Can have specific values based on the Data Range selected in Historical Reporting.

11.3.4.1. Rules applied to determine runtime value

The following rules are used to determine the runtime value. The rules are checked in order, and the first rule to match will set the value:

Rule	Value used
Available Value label matches Data Range	Matching label value
Available Value label does not match Data Range	First Available Value that starts with SELECT
Default	Dataset defined query

11.3.4.2. *Parameter Properties*

Use the following properties for the parameter, where <Dataset Name> is the name of the Dataset for which the parameter applies:

General

Parameter name: report_query_<Dataset Name>

Data type: Text

Allow null value: Checked

Select parameter visibility: Internal

Available Values

Select from one of the following options: Specify values

Add value:

Label	Value
Default	Default

Add at least one of the following:

Label	Value
Interval	Query to use when report interval is Interval
Daily	Query to use when report interval is Daily
Weekly	Query to use when report interval is Weekly
Monthly	Query to use when report interval is Monthly
Any valid label	Query to use if the labels Interval, Daily, Weekly or Monthly are not found.

Default Values

Select from one of the following options: Specify values

Add value: Default

11.3.5. Gather Required Details

The details gathered from the Dataset will be used in the creation of the runtime query parameters.

11.3.5.1. *Before you begin*

The existing Dataset should be defined and tested.

11.3.5.2. Procedure

1. Append the Dataset name to the runtime query parameter names:

Parameter	Name
Timestamp	report_timestamp_<Dataset Name>
Row Filter	report_rowfilter_<Dataset Name>
Query	report_query_<Dataset Name>

2. Convert the existing Dataset query to a query parameter value
 - a. Copy the text of the Dataset query and paste it into a text editor application (i.e. Notepad).
 - b. If the report is grouping or sorting the data, it is recommended to remove the ORDER BY clause.
 - c. Remove items from the query, which are filters for design purposes only. For example:
 - i. TOP Number in the SELECT clause
 - ii. WHERE clause conditions, typically related to timestamp fields
 - d. If a WHERE clause condition remains, it is a runtime condition and will be used as input for the row filter parameter. The remaining conditions are entered as an Available Value for the row filter parameter as follows:

Label	Value
Default	WHERE clause conditions

If no remaining WHERE clause conditions, add the following Available Value for the row filter parameter:

Label	Value
Default	Default

- e. Replace the WHERE clause with the formatted row filter name:
 {?report_rowfilter_<Dataset Name>}

Where <Dataset Name> is the name of the Dataset for which the row filter parameter applies.

11.3.6. Example: Activity Code Statistics Query

This example highlights the process of converting a Dataset query into runtime query parameters.

11.3.6.1. Before you begin

The Dataset query for this example is the query from [Updated Activity Code Statistics Example](#) in [Appendix B: Writing the Report Query](#).

This example is based on the following details:



Dataset name is: dActivityCodeStat

Report will be imported as: Daily Historical report

Existing Dataset query:

```
SELECT TOP 100
MIN(dActivityCodeStat.Timestamp) AS Timestamp,
MIN(LTRIM(dActivityCodeStat.ActivityCode)) AS
ActivityCode,
MIN(LTRIM(dActivityCodeStat.ActivityCodeName)) AS
ActivityCodeName,
MIN(LTRIM(dActivityCodeStat.AgentLogin)) AS AgentLogin,
MIN(LTRIM(dActivityCodeStat.AgentGivenName)) AS
AgentGivenName,
MIN(LTRIM(dActivityCodeStat.AgentSurName)) AS
AgentSurName,
SUM(dActivityCodeStat.ActivityTime) AS ActivityTime,
SUM(dActivityCodeStat.Occurrences) AS Occurrences
FROM dbo.dActivityCodeStat dActivityCodeStat
WHERE (dActivityCodeStat.Timestamp>= DateAdd('d', -8,
Now())) And dActivityCodeStat.Timestamp<= DateAdd('d', -1,
Now())) AND
(dActivityCodeStat.Application<>'System_Application')
GROUP BY dActivityCodeStat.Timestamp,
dActivityCodeStat.ActivityCode,
dActivityCodeStat.AgentLogin
ORDER BY dActivityCodeStat.Timestamp,
dActivityCodeStat.AgentLogin,
dActivityCodeStat.ActivityCode
```

11.3.6.2. *Gather Required Details*

1. Append the Dataset name to the runtime query parameter names:

Parameter	Name
Timestamp	report_timestamp_dActivityCodeStat
Row Filter	report_rowfilter_dActivityCodeStat
Query	report_query_dActivityCodeStat

2. Convert the existing Dataset query to a query parameter value
 - a. Copy the text of the Dataset query and paste it into a text editor application (i.e. Notepad).

- b. The report using this query will be grouped and sorted. Remove the ORDER BY clause will be removed.
- c. Remove the following items from the query, which are filters for design purposes only:
 - i. TOP 100 in the SELECT clause
 - ii. (dActivityCodeStat.Timestamp>= DateAdd('d', -8, Now())) And dActivityCodeStat.Timestamp<= DateAdd('d', -1, Now())) in the WHERE clause
- d. WHERE clause condition remains, so the condition will be entered as an Available Value for the row filter parameter:

Label	Value
Default	WHERE dActivityCodeStat.Application<>'System_Application'

- e. Replace the WHERE clause with the formatted row filter name:
{?report_rowfilter_dActivityCodeStat}

The query produced from this process is:

```

SELECT
MIN(dActivityCodeStat.Timestamp) AS Timestamp,
MIN(LTRIM(dActivityCodeStat.ActivityCode)) AS
ActivityCode,
MIN(LTRIM(dActivityCodeStat.ActivityCodeName)) AS
ActivityCodeName,
MIN(LTRIM(dActivityCodeStat.AgentLogin)) AS
AgentLogin,
MIN(LTRIM(dActivityCodeStat.AgentGivenName)) AS
AgentGivenName,
MIN(LTRIM(dActivityCodeStat.AgentSurName)) AS
AgentSurName,
SUM(dActivityCodeStat.ActivityTime) AS ActivityTime,
SUM(dActivityCodeStat.Occurrences) AS Occurrences
FROM dbo.dActivityCodeStat dActivityCodeStat
{?report_rowfilter_dActivityCodeStat}
GROUP BY dActivityCodeStat.Timestamp,
dActivityCodeStat.ActivityCode,
dActivityCodeStat.AgentLogin
  
```

11.3.6.3. *Timestamp Parameter Properties*

Use the following properties for the timestamp parameter:

General

Parameter name: report_timestamp_dActivityCodeStat

Data type: Text

Allow null value: Checked

Select parameter visibility: Internal

Available Values

Select from one of the following options: Specify values

Add values:

Label	Value
None	(Null)
Daily	Timestamp

Default Values

Select from one of the following options: Specify values

Add value: (Null)

11.3.6.4. *Row Filter Parameter Properties*

Use the following properties for the row filter parameter:

General

Parameter name: report_rowfilter_dActivityCodeStat

Data type: Text

Allow null value: Checked

Select parameter visibility: Internal

Available Values



Select from one of the following options: Specify values

Add values:

Label	Value
TableAlias	*
FieldName	*
Default	WHERE dActivityCodeStat.Application<>'System_Application'

Default Values

Select from one of the following options: Specify values

Add value: Select one of the available values

11.3.6.5. Query Parameter Properties

Use the following properties for the query parameter:

General

Parameter name: report_query_dActivityCodeStat

Data type: Text

Allow null value: Checked

Select parameter visibility: Internal

Available Values

Select from one of the following options: Specify values

Add values:

Label	Value
Default	Default
Daily	SELECT MIN(dActivityCodeStat.Timestamp) AS Timestamp, MIN(LTRIM(dActivityCodeStat.ActivityCode)) AS ActivityCode, MIN(LTRIM(dActivityCodeStat.ActivityCodeName)) AS ActivityCodeName, MIN(LTRIM(dActivityCodeStat.AgentLogin)) AS

Label	Value
	<pre> AgentLogin, MIN(LTRIM(dActivityCodeStat.AgentGivenName)) AS AgentGivenName, MIN(LTRIM(dActivityCodeStat.AgentSurName)) AS AgentSurName, SUM(dActivityCodeStat.ActivityTime) AS ActivityTime, SUM(dActivityCodeStat.Occurrences) AS Occurrences FROM dbo.dActivityCodeStat dActivityCodeStat {?report_rowfilter_dActivityCodeStat} GROUP BY dActivityCodeStat.Timestamp, dActivityCodeStat.ActivityCode, dActivityCodeStat.AgentLogin </pre>

Default Values

Select from one of the following options: Specify values

Add value: Default

12. Appendix F: Import

The ACCS Historical Reporting import feature allows the user to import custom reports. The import provides general validation of the file and user input. If the required contents of the file are not available, the file can still be imported but will fail to run within the Historical reporting application.

12.1. Report is based on the following data

The choice made here determines if Historical Reporting offers the Data Range filter to the user. The choices are:

- Configuration – Does not offer Data Range filtering
- Historical – Offers Data Range filtering

If the report has at least one Timestamp (DateTime) field, the expectation is the report will be imported to allow the user to control the filtering using the Data Range. If the report directly includes all of the required filtering, then the report can be imported as a Configuration report to remove the ability to select the Data Range filtering.

For example, if the report is designed with a filter that returns the previous 7 days of data, and the user does not need control over this range, then it can be imported as a Configuration report.

12.1.1. Runtime query parameters

The Data Range filtering will only work if the report uses runtime query parameters. Runtime query parameters allow the report to have a different runtime query than the original design time Dataset query. See [Parameter Properties](#)

[Use](#) the following properties when adding a runtime parameter, where <Name> is the name of the runtime parameter and <Valid Value> is a value used while designing the report and fit the valid values for the runtime parameter:

General

Parameter name: <Name>

Data type: Text

Allow null value: Checked

Select parameter visibility: Internal

Default Values

Select from one of the following options: Specify values

Add value: <Valid Value>

Runtime Query Parameters in the [Appendix E: ACCS Integration](#) section.

12.2. Import Validation

The import does general validation of the selected file and will only allow the file to be imported if:

- The file extension is RDL or RDLC
- The file is defined using RDL 2010 and is valid based on the RDL 2010 specifications

12.2.1. File Extension Invalid

If the file selected for import does not have the extension RDL or RDLC, the file cannot be imported. The error message displayed to the user is:

Selected file cannot be imported. Please select a Microsoft Report file (*.rdl or *.rdlc).

12.2.2. File is not RDL 2010 or invalid

If the file selected uses a version of RDL other than 2010, the import will fail. The error message displayed to the user is:

Failed to open the report: <File upload path>

Where <File upload path> is the full path of the file selected or entered under **Select Your Report Template**

12.2.3. Report does not have a Timestamp field

If the report selected does not have a Timestamp (DateTime) field and the choice selected for **Report is based on the following data** is Historical, the import will fail. The error message displayed to the user is:

Failed to add new template: <File upload path>

Cannot import this configuration report template

Where <File upload path> is the full path of the file selected or entered under **Select Your Report Template**

12.2.4. Timestamp field not found

If the values enter for the **Database alias and Timestamp field** are incorrect, the import will fail. The error message displayed to the user is:

Failed to add new template: <File upload path>

Timestamp field entered is not a valid field.

Database alias and Timestamp field: <List of valid Timestamp fields>

Where <File upload path> is the full path of the file selected or entered under **Select Your Report Template**, and <List of valid Timestamp fields> is a comma separated list of values that can be used for alias and Timestamp field.

12.3. Historical Reporting Validation

The Historical Reporting validation ensures that the file contains required details. In addition if the report is defined to use the ACCS Data Partitioning feature, Historical Reporting will check that the user has the correct Data Partitioning before running the report.

12.3.1. Report does not include CCMS or CCMM Dataset

The report must have a Dataset using a CCMS or CCMM Data Source or the report will not work in ACCS. The report can be imported; however when selected in Historical Reporting the following error is displayed:

Check if there is a server of type <DSN Prefix> associated with the server <Server Name> and an ODBC DSN created for that server.

Where <DSN Prefix> is the DSN name to the first underscore character, and <Server Name> is the server for which the report was imported.

12.3.2. Data Partition

If the report Dataset is associated with ACCS data partitioning, and the current user does not have the required data partitioning elements defined the report cannot be used. The error message displayed to the user is:

You cannot run/submit this report because the partition assigned to you does not contain the data elements required for this report.

13. Appendix G: Collections

The report uses collections. The collections available are:

Collection	Description
Globals	Collection of global variables related to the report.
User	Collection of data about the user running the report.
Parameters	Collection of report parameters.
Fields	Collection of fields that are available to the report
DataSets	Collection of datasets referenced from the body of a report definition.
DataSources	Collection of data sources referenced from within the body of a report.
Variables	Collection of report variables and group variables.
ReportItems	Collection of text boxes for a report item.

13.1. Global Collection

Collection	Item	ACCS Supported	Description
Globals	ExecutionTime	Yes	The date and time that the report began to run.
Globals	OverallPageNumber	Yes	The page number of the current page for the entire report.
Globals	OverallTotalPages	Yes	The total number pages for the entire report.
Globals	PageName	Yes	The name of the page. The initial value is set using the Report property InitialPageName. As each report item is processed, this value is replaced by the corresponding value of PageName.
Globals	PageNumber	Yes	The current page number relative to page breaks that reset the page number.
Globals	RenderFormat.IsInteractive	Yes	Whether the current rendering request uses an interactive rendering format.
Globals	RenderFormat.Name	Yes	The name of the current renderer.
Globals	ReportFolder	No	The full path to the folder containing the report. This does not include the report server URL.
Globals	ReportName	No	The name of the report as it is stored in the report server database.
Globals	ReportServerURL	No	The URL of the report server on which the report is being run.
Globals	TotalPages	Yes	The total number of pages relative to page breaks that reset PageNumber.

13.1.1. RenderFormat

The RenderFormat object in the Global collection can be used to conditionally format items based on the current target. The following chart shows the values for each target:

Target	RenderFormat.Name	RenderFormat.IsInteractive
Report Viewer	RPL	True
Adobe Acrobat PDF	PDF	No
Excel	EXCELOPENXML	No
Word	WORDOPENXML	No
Print	IMAGE	No

The values can be used in expressions to control the report based on the render format. For example the following expression returns True when the report is exporting to Excel:

```
Globals!RenderFormat.Name = "EXCELOPENXML"
```

13.2. User Collection

Collection	Item	ACCS Supported	Description
User	Language	Yes	The current page number relative to page breaks that reset the page number.
User	UserID	No	The ID of the user running the report.

14. Appendix H: Globalization

When formatting a value, the preference is to use the Microsoft standard format strings. These are a single character, optionally followed by a numeric precision specifier, which use the user's locale to determine the format.

The following table shows a few of the format strings and the result based on the user's locale.

Type	Format String	Name	Locale: Example Value -> Result
Numeric	N	Number	en-US: 1234.567 -> 1,234.57 ru-RU: 1234.567 -> 1 234,57
Numeric	N1	Number	en-US: 1234 -> 1,234.0 ru-RU: 1234 -> 1 234,0
Numeric	P	Percent	en-US: 1 -> 100.00 % fr-FR: 1 -> 100,00 %
Numeric	P1	Percent	en-US: 0.39678 -> 39.7 % fr-FR: 0.39678 -> 39,7 %
DateTime	d	Short Date	en-US: 2014-06-15 13:45:30 -> 6/15/2014 en-UK: 2014-06-15 13:45:30 -> 15/06/2014
DateTime	t	Short Time	en-US: 2014-06-15 13:45:30 -> 1:45 PM en-UK: 2014-06-15 13:45:30 -> 13:45
DateTime	T	Long Time	en-US: 2014-06-15 13:45:30 -> 1:45:30 PM en-UK: 2014-06-15 13:45:30 -> 13:45:30

The following links provide a complete list of the standard format strings:

Numeric: [http://msdn.microsoft.com/en-us/library/dwhawy9k\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/dwhawy9k(v=vs.100).aspx)

Date Time: [http://msdn.microsoft.com/en-us/library/az4se3k1\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/az4se3k1(v=vs.100).aspx)

Timespan: [http://msdn.microsoft.com/en-us/library/ee372286\(v=vs.100\).aspx](http://msdn.microsoft.com/en-us/library/ee372286(v=vs.100).aspx)

14.1. Format using the Properties dialog

The format of a value can be set in the objects Properties under Number. Number is a general name for formatting and is not limited to numeric formatting.

To ensure that the user's preferences are used when formatting values:

Format	Select
Number	Number category and check Use regional formatting

Format	Select
Date	Date category and formats that begin with an asterisk (*) under Type
Time	Time category and formats that begin with an asterisk (*) under Type
Percentage	Custom category and enter p for Custom format

14.1.1. Confirming the format

To confirm if the format used is a Microsoft standard format strings, or to see the format string:

1. In the **Properties** dialog select **Number**.
2. Select the **Category** to use for formatting the value.
3. Adjust other settings as required.
4. Select the **Custom** category
5. The current format string is displayed under **Custom format**.

15. Appendix I: Frequently Asked Questions

15.1. Page Breaks

15.1.1. How are page breaks set

To separate the content of a report into individual pages, use a logical page break. Logical page breaks are defined as a property in various report items, including group, rectangle, list, table, matrix, and chart.

15.1.2. When can a page break occur

The supported page break locations include:

- Page break before (Start)
- Page break after (End)
- Page break before and after (StartAndEnd)

15.1.3. How can page breaks be set if not support by item

If the report item does not offer page break, the item can be inserted inside a report item that does support page breaks. For example, the rectangle object can be used to enforce a page break for objects placed in the rectangle.

15.1.4. How can a conditional page break be defined

The page break location cannot be conditionally set since the property does not support the use of expressions. Instead, the selected page break location can be disabled based on an expression, using the **Disabled** property under **Page Break**.

15.2. Charts

15.2.1. Why do charts change position on report

Charts can grow in height and width based on the data selected for the chart. SSRS makes its best guess at aligning objects in reports, but sometimes, especially when one element is fixed size and the other is variable, this doesn't always work.

15.2.2. How to stop charts from changing position

The easiest way to stop charts changing position is to add a Rectangle item to the report, and move the chart into the rectangle. Position the rectangle where the chart should appear on the report. Resize the rectangle to the minimum size for the chart. When done correctly. The Parent attribute of the chart will change to the name of the Rectangle object.

15.2.3. Why do charts appear blank

The chart can appear blank even if there is valid data for the chart. The issue occurs when the charts are placed in groups and sub-group, and document map is used.

To correct the issue, place a rectangle on the report, and then placing the chart in the rectangle.



16. References

16.1. Documentation

The following links contain details on standard documentation related to this application note.

Administrating Avaya Contact Center Select

<http://support.avaya.com/css/P8/documents/100181092>

Avaya Aura Contact Center Performance Management Data Dictionary (44400-117)

<https://downloads.avaya.com/css/P8/documents/100178586>

Avaya Aura Contact Center Server Administration (44400-610)

<https://downloads.avaya.com/css/P8/documents/100178603>

17. Glossary of Terms

AACC	Avaya Aura® Contact Center
ACCS	Avaya Contact Center Select
AML	Application Module Link
CCMA	Contact Center Manager Administration
CCMM	Contact Center MultiMedia
CCMS	Contact Center Manager Server
DSN	Data Source Name
FP	Feature Pack
GA	Generally Available
ODBC	Open Database Connectivity



OS	Operating System
RDL	Report Definition Language
RDLC	Report Definition Language for Client reports
SP	Service Pack
SSRS	SQL Server Reporting Services
SQL	Structured SQL Language

©2015 Avaya Inc. All Rights Reserved.

Avaya and the Avaya Logo are trademarks of Avaya Inc. All trademarks identified by ® and ™ are registered trademarks or trademarks, respectively, of Avaya Inc. All other trademarks are the property of their respective owners. The information provided in these Application Notes is subject to change without notice. The configurations, technical data, and recommendations provided in these Application Notes are believed to be accurate and dependable, but are presented without express or implied warranty. Users are responsible for their application of any products specified in these Application Notes.