**Creating Custom Paginated Reports in MinistryPlatform**

## Background

MinistryPlatform allows for custom reports to be created and launched from within the MP application. Creating reports requires experience in T-SQL programming, creating stored procedures, report design using the MS Visual Studio or MS Report Builder programs, SQL Server Reporting Services administration, and MP administration. It is recommended to become familiar with the many reports MP makes available to users prior to creating a new one. The article describes how to create custom reports in MP.

Report writing in and of itself does not require changes to existing database schema or objects, but is it highly recommended to conduct tests in a test environment before deploying reports to the production server.

The reports discussed in this document refer to paginated reports (not mobile reports).

**What is a Report?**

A paginated report consists of one or more datasets formatted to view as a printed document or a PDF file. Create a report if you need to print a set of data in an easy to read format. If data only needs to be viewed in a grid format and the data does not need to be printed, then create a view (they are A LOT easier to make).

Necessary Elements for Creating Reports  
There are three elements necessary to create a report for MinistryPlatform: a **data source**, a **dataset** (populated with a stored procedure) and one or more **report objects** to display the dataset.

**Data Source** - A data source includes the data source type, connection information, and the type of credentials to use. There are two types of data sources: embedded and shared. An embedded data source is defined in the report and used only by that report. A shared data source is defined independently from a report and can be used by multiple reports. *It is recommended to use a shared data source for reports accessing MP.*

If you choose to use the existing MPReportsDS data source created by ThinkMinistry, note the source type is “OLE DB”. You may find using the “SQL Server” type as a data source will expose more features to the report authoring tool.

**Dataset** - To add data to a report, you create datasets. Each dataset represents the result set from running a query command on a data source. The columns in the result set are the field collection. The rows in the result set are the data. A dataset does not contain the actual data. A dataset contains the information that is needed to retrieve a specific set of data from a data source.

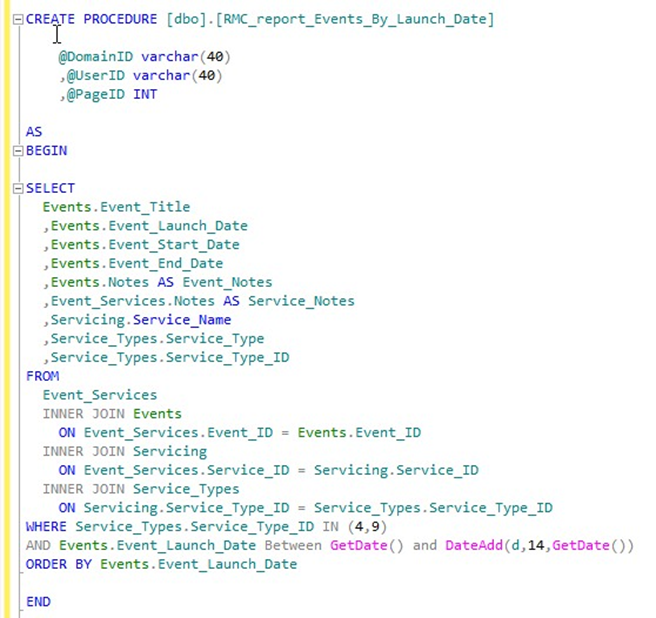
**Stored Procedure** – A saved and compiled T-SQL (Transact-Structured Query Language Statement) statement. SQL is the language used to interact with data in a database. The primary tasks of SQL are to read data, insert data, change data and delete data. Teaching how to write T-SQL code is beyond the scope of this document, but there are many great video tutorials online. See a reference to one such course at the end of this document.

The dataset(s) required for any report within MP must be created using SQL Server stored procedures because MP reports are required to have three input parameters (Domain GUID, the User GUID, and the Page ID).  The Domain GUID can be found in the System Setup section (**System Setup** -> **Domains/Accounts**). A User GUID is a value unique to each user.  A User GUID can be found for a specific user on the **Users** page within the **Administration** section. In general, these parameters are used to check security, restrictions, limit data, etc. within the report. It allows for integration with the security roles set up with MP. The Page ID is the actual numerical ID of the page the report is being launched from.  The Platform will pass the parameters to the stored procedure. The stored procedures need only be declared. *When creating your own stored procedures, it is best to assign a prefix to the stored procedure name to easily identify custom stored procedures from those created by ThinkMinistry.*

Required Input Parameter Declarations for MP Stored Procedures

@DomainID VARCHAR(40)  
@UserID VARCHAR (40)  
@PageID INT

Stored Procedure Example



**Figure 1**

**Report Object** – The object that is placed on a report that displays the report’s dataset in a meaningful way. Examples of an object are a table, matrix or a chart.

## Report Authoring Tools **Microsoft Visual Studio**

Report authoring can only be done in the version of Visual Studio that supports your specific SQL Server installation.  In fact, when you install SQL Server, you have the option of installing a version of Visual Studio for report development.  For example, SQL 2008 installs a shell of VS 2005; SQL2008R2 installs a shell of Visual Studio 2008; SQL 2012 installs a shell of VS 2010; and SQL 2014 installs a shell of VS 2012.  In general, the SQL version is one revision ahead of the VS version you use and the version of VS you get is specifically limited to report authoring.  If you have a full compatible version of Visual Studio you can use your own copy but it's usually a good thing to use the version of VS that comes with SQL.  If you don't use that specific version, things can get pretty hairy.

**Microsoft Report Builder**

Report Builder is a report authoring environment for business users who prefer to work in the Microsoft Office environment. Report Builder is a free download from Microsoft that can be installed on a Windows client or on the database server. For more details, read the Microsoft documentation in the MSDN library.

Unless otherwise noted, all directions and screenshots use the MS Report Builder 3.0 user interface.

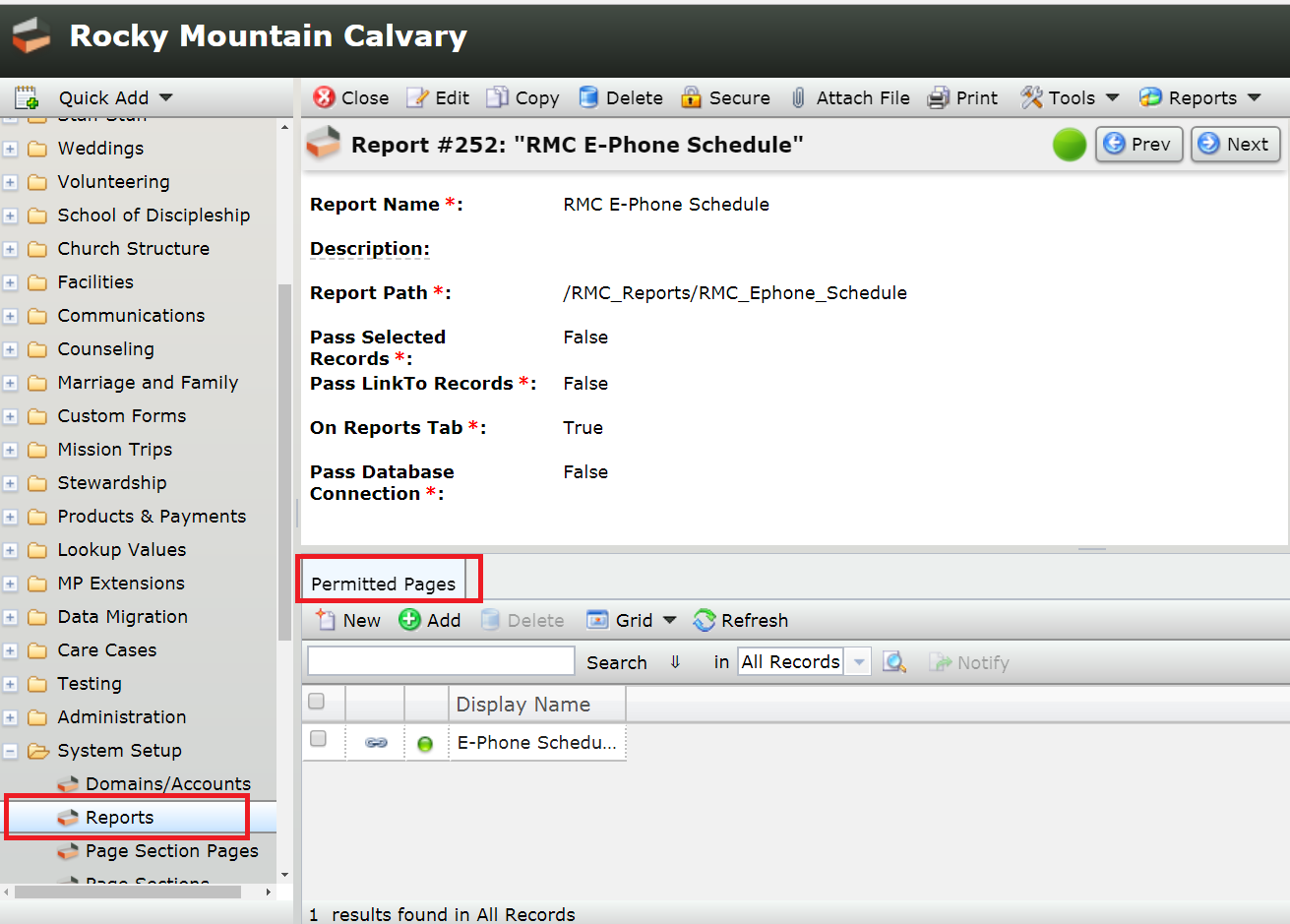
## SQL Server Reporting Services (SSRS)

SSRS is used to manage and store reporting files. This application must be used to manage any reports created with Visual Studio or Report Builder. SSRS can be installed using your SQL Server installation media. The SSRS installation process creates a database on SQL Server for your reports and it also installs a web service/application to manage the report files. SSRS is to your web authoring app what Windows Explorer is to MS Word. In other words, using SSRS you can create folders, rename reports, and manage your report in many other ways. The default path for SSRS is <http://[SQL_Server_Computer_Name]/reports>. For example, if the computer name is **MP**, then the URL would be **http://mp/reports**.

## Deploying Reports to MinistryPlatform

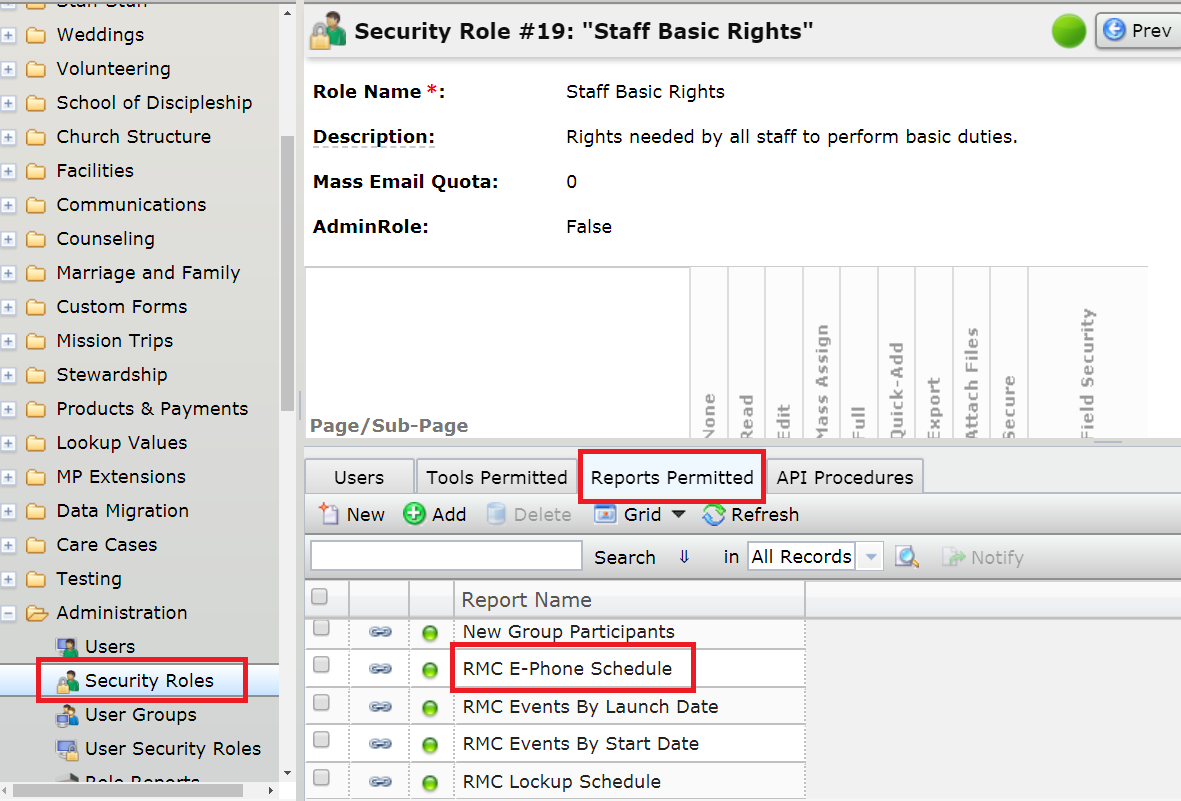
Once a report is created, it must be made visible to MP. The basic concept in report deployment is to create the report object in MP, assign the report to a page or pages, and assign users permission to access the report.

In MP go to the **Reports** page under **System Setup**.  You can create a new report record on the page. Give the report the name and set the report path. The path is just the exact path to the report inside of SSRS with a slash at the beginning.  So, if you create a report titled **RMC\_Ephone\_Schedule** into a folder called **RMC\_Reports**, the report path would be **/RMC\_Reports/RMC\_EPhone\_Schedule** (spaces are permissible—you don’t have to use underscores). At that point, you can add which pages you want into the sub-page of that report and they will show up on that page. See figures 4 and 5.



**Figure 2**

Also understand that the reports are role-based, so you have to go into **Role Reports** under **Administration** and make sure the roles are set correctly.  The roles are the same as the security roles, so all the security is wired in.



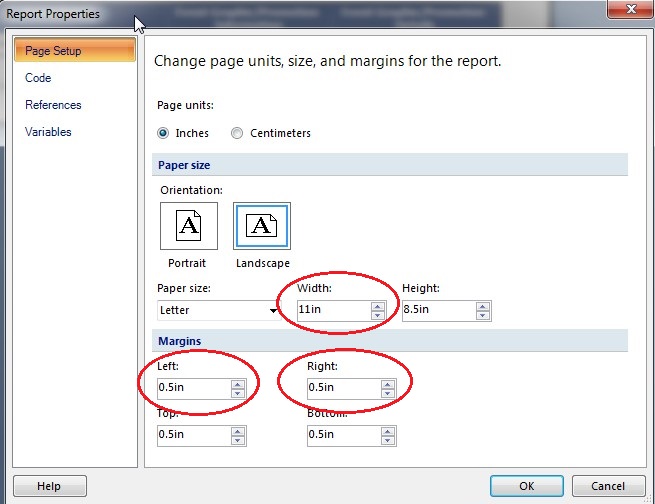
**Figure 3**

## Report Format for Printing

When laying out objects (tables, matrices, charts, textboxes, lines, etc.) using your report authoring tool, you must be careful not to cause the report body to be wider than the width of the paper size of the report. Adobe Reader will not make automatic adjustments to keep the report from being *too wide*. If the overall width is larger than what Adobe Reader is expecting or if there are report object that exceed the page’s wide, Adobe Reader may display blank pages that you do not expect.

One way to ensure this does not happen is to run the report and click the **Print Layout** button. This view will indicate if the report will fit on the page as expected.

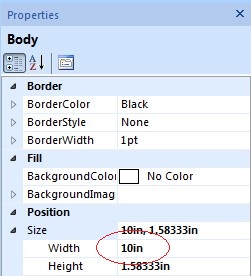
To ensure the report prints properly, open your report in Report Builder. Right-click in the main authoring background area (anywhere other than on the body of the report) and select **Report Properties.** This dialog will allow editing of the paper size, units, orientation and margins (see **Figure 4**). Make note of the paper **width** and **right** and **left** **margins**. In most cases, your width and height will either be 8.5in x 11in (portrait) or 11in x 8.5 in (landscape).



**Figure 4**

Close the **Report Properties** window.

Right-click on any blank area in the body of the report page and find the body’s **Position** property category. Expand **Position** and **Size** and make note of the **Width** property value. See **Figure 5**.



**Figure 5**

Using the values in the screenshots above, note the total report width is 11 in and oriented as a landscape document. The report body width is 10in and the report margins total 1 in (left = 0.5 in., right = 0.5 in.). If total width was anything greater than 11 inches, then the report width would “run over” on to other pages thereby printing blank pages in Adobe Reader.

## Report Creation and Testing

When running reports in development and when deploying them, keep the following practices in mind.

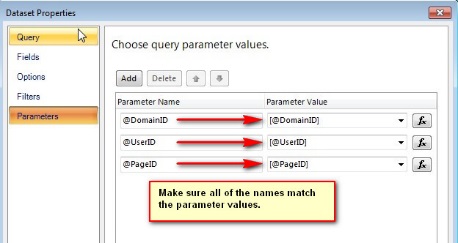
**Running Reports in Development**

During development (i.e., when running report from the report authoring tool) it may be best to use default values for required parameters so values do not have to be entered every time a report is run. To set a default parameter during testing, expand the **Parameters** folder in the **Report Data**. Right-click each parameter and select **Parameter Properties**. Click on the **Default Values** list item on the left and select **Specify values**. Enter the appropriate values for each parameter.

**Preparing Reports for Production**

Prior to placing the report into production remove all parameter values by opening each parameter property (see previous section on how to open each parameter property). Simply open each parameter value and delete it.

Next, verify that each report parameter name is matched to the corresponding stored procedure parameter value. In the **Report Data** window, right-click on the report dataset and select **Dataset Properties**. Click on the **Parameters** list item on the left and match each input parameter with the parameter value of the same name. See Figure 6.



**Figure 6**

**Best Practices**

Report File Storage

In SSRS, be sure to create your own folders to store your report files. This differentiates your report files from those created by other organizations.

Stored Procedure Naming Conventions

When saving stored procedures, create a prefix to keep your stored procedures differentiated from those created by other organizations. For example, for the church name **Rocky Mountain Calvary**, a suggestion is **RMC\_report\_[StoredProcedureName]**.

## Miscellaneous

**Video Tutorials**

The video tutorials below reference the site Lynda.com. Lynda.com is not a free service, but your local library district may have an agreement with Lynda.com to provide free tutorials if you have a library card number and PIN. This site may be helpful in this regard: <https://lifehacker.com/here-s-how-to-access-lynda-s-learning-database-for-free-1820761688>.

|  |  |
| --- | --- |
| **Course Name** | **URL** |
| Learning SQL Programming | <https://www.lynda.com/SQL-tutorials/Learning-SQL-Programming/548044-2.html> |
| SQL Server Reporting Services In Depth | <http://www.lynda.com/SQL-Server-tutorials/SQL-Server-Reporting-Services-Depth/110282-2.html> |

**Report Definition Files (RDL)**

A Report Definition Language (RDL) file is created by SQL Server Reporting Services when using one of the aforementioned report authoring tools. A report definition contains data retrieval and layout information for a report.

RDL files are managed through the SQL Server Reporting Services (SSRS) web interface which you can get to by being on the network or on the server running SSRS.  It's usually http://<servername>/Reports.

The RDL can be created within any directory inside the SSRS web interface, but it's probably best to create your own directory apart from the MinistryPlatform directories since MP reports are often created or updated.

In summary, RDL files are created by a report authoring tool and saved and managed from within SSRS (http://<servername>/Reports). Note, when opening the SSRS web application, the web browser may need to be run as an administrator.