Custom Reporting Guide

Version 4.0

Updated May, 2012

Table of Contents

Overview

Notes for this Document

- **This document** is intended for RMTrack administrators and report developers. RMTrack also publishes an Administration Guide for site administrators and a User Guide for general users. However, any RMTrack user may benefit from reading all of the guides as certain features and functionality are explained in only one of the guides.

- **Help pages** are available within the application on every RMTrack page by clicking on the Help link located at the top left of each page. The User, Administration and Reporting guides are also available on-line from Help by selecting Help → Contents → Documents.

- **Issues** and **Projects** are two terms used throughout this document. Issue refers to the detail level of information. An issue can be thought of as one item or record within the application. Project refers to a group of issues linked by some common factor, for example, issues for a certain client or issues for a certain product. One of the features of RMTrack is the ability to change the naming of Issue and Project. For example the name for Issues might be changed to Defects, Bugs or Tasks and Projects might be changed to Clients or Products. The decision and ability to change these names is made by the site administrator. This document does not reflect any custom name changes made to a specific installation. Therefore, users may not see the word Issue or Project in their installation.

- **Issue Numbering** is another configurable feature of RMTrack. By default RMTrack numbers each issue uniquely across all projects i.e. Issue number 1 will only appear in one project. This is the issue numbering style presented in this document. Site administrators may choose to alter the issue numbering to 'sequentially within each project' i.e. each project will have an issue 1, for example P1-1 and P2-1 where P1 and P2 are project prefixes assigned by the administrator.

- **Please report** any errors or discrepancies you may find in this document to support@RMTrack.com. Your assistance is appreciated.

New with V4.0

Features and Usability

Complete lists of bug fixes, usability changes and new features are available from: **Help→Contents→ReadMe** under the New Features & Enhancements section.

Release notes for all versions and releases of RMTrack are also available from the download page of our website: *www.rmtrack.com/downloads*.

The following list contains some new feature and usability additions of special interest in V4.0 and where to find them in this document.

New Features Reference List

Getting Started with Custom Reporting

RMTrack allows clients to create and publish their own reports by writing SQL queries. Using this feature requires both knowledge of SQL and knowledge of the RMTrack database structure.

A Note from RMTrack Support

RMTrack offers full pre and post-sale support for our application. We are happy to answer any questions you have and offer any assistance required to introduce you to the application and to assist with configuring RMTrack to meet your specific needs.

RMTrack Support has years of experience in a variety of businesses and have assisted in the configuration of hundreds of RMTrack installations.

Please call or e-mail RMTrack Support to request assistance, guidance, online meetings … anything you need to get the most from RMTrack.

Activating the Custom Reporting Feature

Custom Reports allows clients to create SQL scripts within the application that will call the SQL Server to generate the report. It is possible to write “reports” that can actually update the database (or in a less secure environment – theoretically update any database on the server).

> **WARNING: We highly recommend that you involve your database/network professionals when activating the Custom Reports feature.**

RMTrack uses a database connection file called DBConnection.udl to tell the application how to access the RMTrack database. This file is stored in your RMTrack virtual directory (default location is: C:\Inetpub\wwwroot\RMT).

To activate Custom Reports a new and separate database connection file is required. This new file must be called **DbConnectionReporting.udl** and must reside in the sub-folder called **ReportingConnections**.

> **WARNING: It is strongly recommended that this connection file specifies a database user that has reporting capability ONLY. This will eliminate the possibility of the RMTrack database (or any other on the server) from being updated via the custom reporting feature.**

Each individual report can also have its own database connection file, allowing different levels of database access for each report. You can create separate database connection files that have the same name as the report e.g. **MYREPORT.UDL.** When the application attempts to run a report, it will first check to see if the report has its own unique database connection file. If the report has its own database connection files then it will be used to connect to the database. If a report does not have its own database connection file the report will use the default DbConnectionReporting.udl.

Basics of Reporting

When designing any report there are a number of questions that should be asked:

- What is the purpose of this report?
  Writing down the purpose of a new report allows you to focus on exactly what data should and should not be in the report. Writing a good goal or purpose for a report lets know where you're going with the report.

- Who is the intended audience of the report?
  Make sure these users are allowed to access this data. Does the report contain confidential information?

- What tables and fields do you want to include in the report?
  Before you start your report, you should know the exact tables and fields that will be in the report.

- How do you want the data grouped? What summary calculations do you need?
  Grouping and calculating in SQL can be complicated and you need to make sure you have a firm understanding of exactly how to accomplish your grouping and summation needs.

- Do the report users need to change the query?
  A report can be much more useful if the user can specify different parameters, like a date range. Planning for this at the outset will make the whole process much easier.

Sample Reports

You can download a set of example reports from the Downloads page of the RMTrack website or directly from http://www.rmtrack.com/downloads/SampleReports.zip.

These will assist by providing a basic template for various reports. To install these reports:

1. Download, unzip and save the sample reports (above) somewhere accessible.
2. Go to **Administration→Customize →Custom Reports**
3. Click the **Import…** button
4. Browse to the sample report files and import them one by one.
   NOTE: you must import the reports named *Example - Dashboard* and *Example - Hyperlinks* last as they contain sub-reports which must be loaded first.

When reports are imported, the publishing rules are set to **Hidden**. To publish these reports:

1. Open the report by clicking on the Report Name.
2. In the first section, set **Publish report to:** to **Site Administrators**
3. Click **Save**

As each report is published you should see it appear in the left side menu under the **Reports** heading.

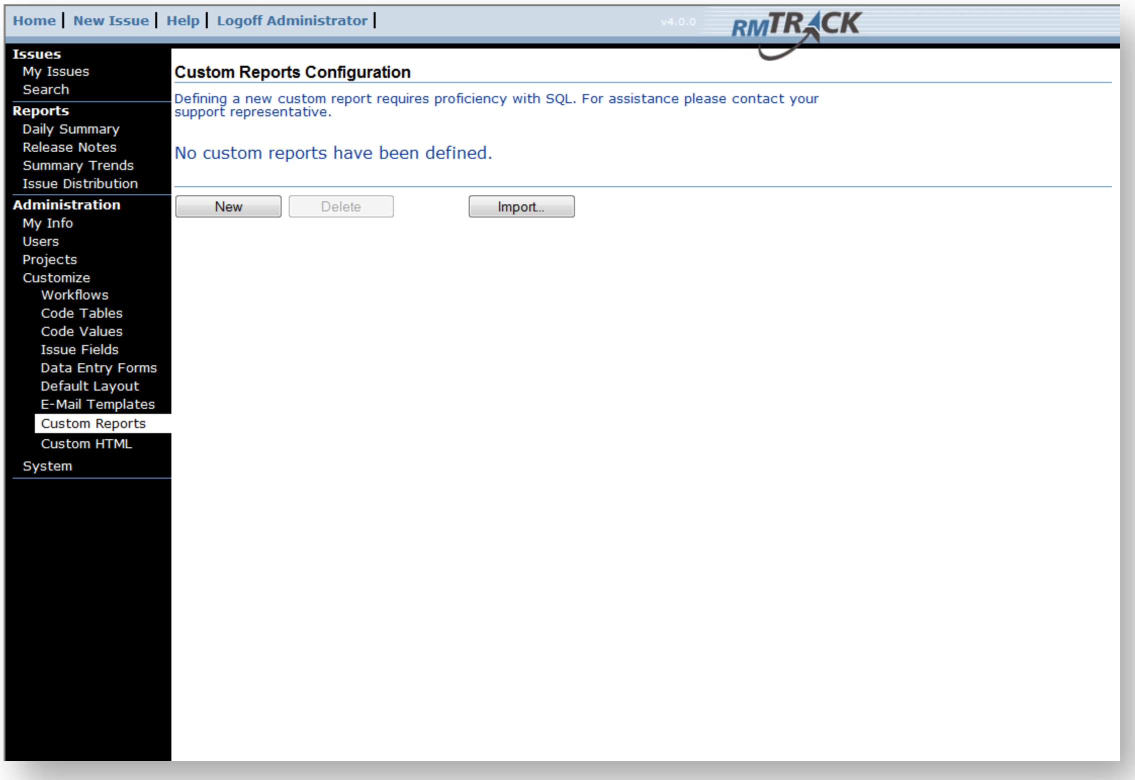
Note that these reports are designed to work with any installation by only using fields that are available in all RMTrack databases. However, this means that these reports may not show very *meaningful* results. They are designed as an introduction and example of different types of reporting only.

The reports that you are importing are:

- **Average Time Held** – Calculates the average amount of time that an issue spends in each resolution state. This is a complex report that makes use of a Project parameter, a “temp” table, a, multiple “joined” tables
- **Dashboard** – This is an example of combining multiple reports on one page and uses four of the other example reports to produce a *dashboard* effect. This *parent* report uses *Issue by Project, Project Parameter - bar chart, Top Ten Oldest,* and *Summary Trends* as sub-reports.
- **History Details** – Displays the user, date, From Resolution and To Resolution for an issue for every time the resolution changes. This report is used as an example of a *drill-down* report accessed from a hyperlink in a *parent* report.
- **Hyperlinks** – Provides examples of using hyperlinks in reports. This report uses 4 types of hyperlinks: Link to Issues Details page, Link to a *drill-down* sub-report, link to e-mail client, link to an external URL.
- **Issues by Project** – A simple report that graphs number of open issues by project.
- **Project Parameter** – A simple example of a report that uses a parameter (project) and counts issues by user id.
- **Project Parameter (Bar chart)** – Same as the Project Parameter report except that the results are displayed as a bar chart.
- **Simple Query** – A very simple report that lists all Issue Id’s
- **Summary Trends** – Basically a copy of the “stock” Summary Trends report. This is another example of a very complex report.
- **Top Ten Oldest** – This report lists the ten oldest open issues.

Creating and Modifying Reports

To view or modify an existing report, click its name. To add a new report, click the **New** button. To delete a Custom Report, check the box to the left of the name and then click the **Delete** button.

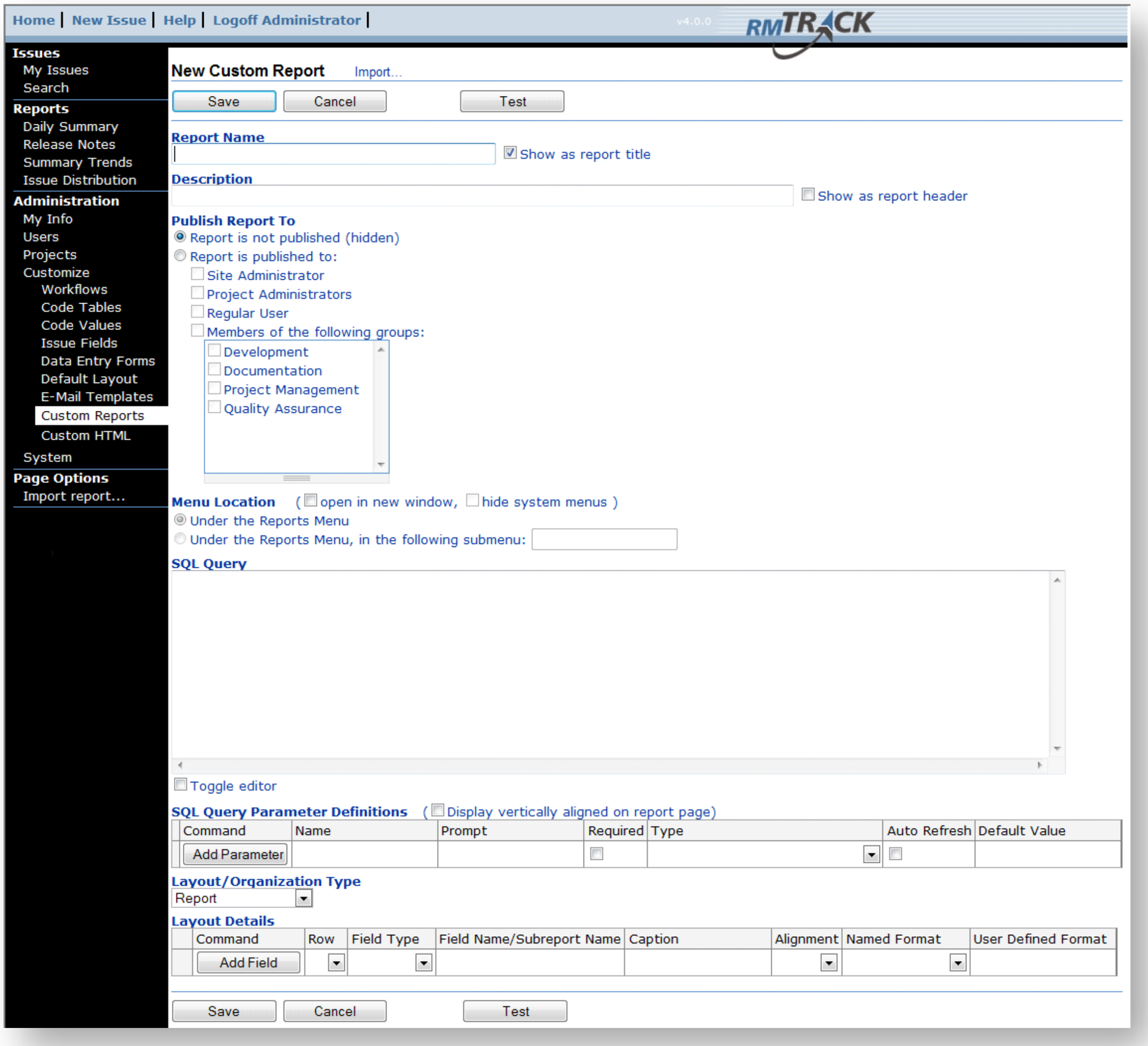


Custom Reports Configuration

Defining a new custom report requires proficiency with SQL. For assistance please contact your support representative.

No custom reports have been defined.

RMTrack Custom reports can be imported and exported between RMTrack Installations. To import a report click the **Import…** button, browse to the report file and then click the Import button.



New Custom Report Import...

Report Name

☑ Show as report title

Description

□ Show as report header

Publish Report To

● Report is not published (hidden)

○ Report is published to:

□ Site Administrator

□ Project Administrators

□ Regular User

□ Members of the following groups:

□ Development

□ Documentation

□ Project Management

□ Quality Assurance

Menu Location (□ open in new window, □ hide system menus)

● Under the Reports Menu

○ Under the Reports Menu, in the following submenu:

SQL Query

□ Toggle editor

SQL Query Parameter Definitions (□ Display vertically aligned on report page)

Command	Name	Prompt	Required	Type	Auto Refresh	Default Value
Add Parameter			□	▾	□	

Layout/Organization Type

Report ▾

Layout Details

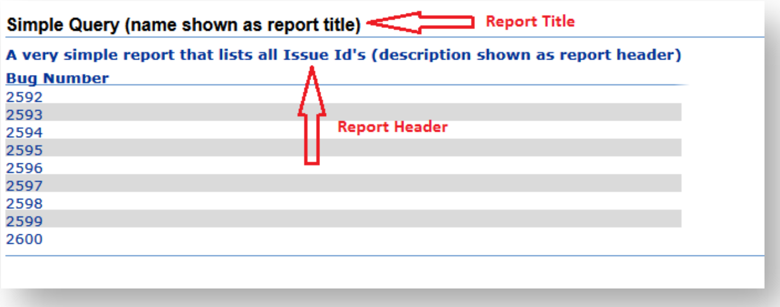
Command	Row	Field Type	Field Name/Subreport Name	Caption	Alignment	Named Format	User Defined Format
Add Field	▾	▾			▾	▾	

Report Name and Description

The **Report Name** is what will appear in the left side RMTrack menu for any users that have access to the report. This name should be meaningful enough to identify the report without being excessively long as this will cause your left side menu bar to grow wider.

The **Show as report title** checkbox controls whether the Report Name will be shown as a title on the page when the report is run.

The **Description** will appear beside the report name in the list of reports and can also be displayed as part of the report by checking the **Show as report header** box.



Simple Query (name shown as report title)

A very simple report that lists all Issue Id's (description shown as report header)

Bug Number
2592
2593
2594
2595
2596
2597
2598
2599
2600

Report Publishing

You can choose who can see and run a report. Users that cannot access a report will not see the report listed in the left menu

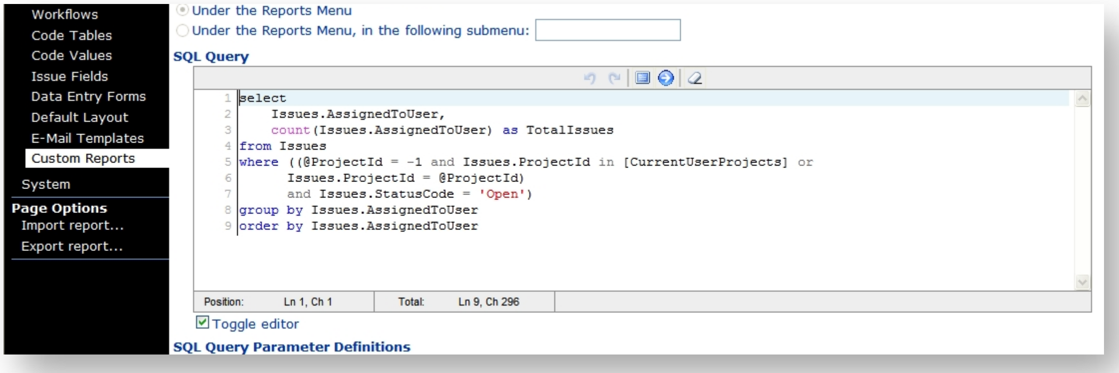
Reports can be published by **User Type** (e.g. Site Administrators, Project Administrators, Regular Users, Public Users) and by **Group Membership** by checking the appropriate boxes in the **Publish Report To** section.

Reports can also be **Hidden**. Some reports may be specifically designed as **Sub-reports** and have little use on their own in which case you may choose to hide the report so that it can only be viewed by running the parent report. You may also want the report to not show up under the **Reports** section of the left side menu but somewhere else in the menu. This can be accomplished by setting the report to **hidden** using the report publishing but then adding the report as a **Custom Menu Entry**. For more information on **Menu Entries** please see the Administration Guide.

The **Open in new window** checkbox controls whether the report will use the same browser window when run or will open a new browser window. If you chose to have the report open in a new window you can also choose to have the report display without the RMTrack menus at the top and the left side by using the **hide system menus** option.

Reports published here can be shown under the general **Reports** menu heading or you can create a submenu to group reports under. For example you may want to group some reports under a submenu of **Productivity Reports.** Each report that uses the same submenu name will appear under that heading in the Reports section.

SQL Queries



● Under the Reports Menu
○ Under the Reports Menu, in the following submenu:

SQL Query

```
select
    Issues.AssignedToUser,
    count(Issues.AssignedToUser) as TotalIssues
from Issues
where ((@ProjectId = -1 and Issues.ProjectId in [CurrentUserProjects] or
      Issues.ProjectId = @ProjectId)
      and Issues.StatusCode = 'Open')
group by Issues.AssignedToUser
order by Issues.AssignedToUser
```

Position: Ln 1, Ch 1 Total: Ln 9, Ch 296

☑ Toggle editor

SQL Query Parameter Definitions

The actual SQL query to generate the data for the report is entered into the **SQL Query** area.

There is a checkbox below the Query area that allows you toggle between a simple text entry and a more robust editor. This editor has options for:

Full-screen mode

Undo/redo

Go to line #.

Basic Querying

SQL is a very powerful and complex programming language that can produce virtually any result desired. Constructing an SQL query requires considerable knowledge of SQL and of the data to be reported on.

An SQL query for an RMTrack custom report must produce a single **result set**, i.e. the SQL must return the results of a single SQL select statement. Multiple result sets are not supported.

The following example returns a single result set (a list of all issues):

```
SELECT * FROM Issues
```

An example of a multiple result sets, which are not supported (a list of all issues, followed by a list of all projects):

```
SELECT * FROM Issues
SELECT * FROM Projects
```

Advanced Querying

SQL is an exceptionally powerful language and a query for a custom report is not restricted to just the SELECT statement. As long as the query eventually returns a single result set, the query can consist of just about anything.

A common approach is to use a temporary table to produce interim results and at the end return the entire temporary table as the result set. For example:

```
--
-- Define the temporary table to hold the calculations
--
if object_id('tempdb..#SummaryData') is null begin
    create table #SummaryData
    (
        ActivityDate datetime primary key,
        TotalIssues int,
    )
end

--
-- Record dates issues were opened, and how many
--
insert into #SummaryData
select
    distinct(DateOpened) as ActivityDate,
    count(*) as TotalIssues
from Issues
group by Issues.DateOpened
order by Issues.DateOpened

--
-- Return the data
--
select * from #SummaryData order by ActivityDate

--
-- Done with the temporary table
--
drop table #SummaryData
```

Issue Row Level Security

RMTrack provides considerable security in terms of which users can access which issues. When writing an SQL query for a custom report it is strongly recommended that the queries provide the same level of security. To help with this several **macros** have been defined. Whenever one of these macros is encountered in a query it is substituted with the appropriate SQL.

The following macros are supported:

Macro	Purpose

[CurrentUserProjects]	A parenthesized list of ProjectId's the user executing the report belongs to. For example: (1, 2, 4). This macro is intended for use with SQL's "IN" operator, to ensure that the query only returns data for projects the user belongs to. For example:
	SELECT * FROM [Issues] WHERE Issues.ProjectId in [CurrentUserProjects]
	The query actually executed will be:
	SELECT * FROM Issues WHERE Issues.ProjectId in (1, 2, 4)

[IssuePreFilterAnd]	The SQL expression that is defined by the Site Option PreFilter, ending with an AND. This macro is intended to ensure that any issue prefilter conditions can also be applied. For example: SELECT * FROM Issues WHERE [IssuePreFilterAnd] Issues.ProjectId in [CurrentUserProjects] If a pre-filter has not been defined then this macro is an empty string.
[AndIssuePreFilter]	The SQL expression that is defined by the Site Option PreFilter, starting with an AND. This macro is intended to ensure that any issue prefilter conditions can also be applied. For example: SELECT * FROM Issues WHERE Issues.ProjectId in [CurrentUserProjects] [AndIssuePreFilter] If a pre-filter has not been defined then this macro is an empty string.
[IssuePreFilter]	The SQL expression that is defined by the Site Option PreFilter, without an AND at the start. This macro is less useful than the [AndIssuePreFilter] because to correctly use it the query author needs to know that a prefilter has been defined. If a pre-filter has not been defined then this macro is an empty string.
[IssueRowLevelSecurity]	An SQL expression that exerts row level security constraints against a query on the Issues table. See below for more information.

RMTrack's full issue level security consists of the following requirements:

- Site administrators can access any issue
- A user that is member of at least one group with unrestricted access can access any issue in a project they belong to
    - Further restricted by the issue pre-filter site option
- A user that is not member of any group with unrestricted access can only access issues they have created, are currently assigned to, or have been assigned to at some point in the issue's history
    - Further restricted by the issue pre-filter site option

These requirements have been wrapped up in a macro called [IssueRowLevelSecurity] which evaluates to the following SQL expression:

```
(exists(select * from Users
        where Users.UserId = @CurrentUserId
        and Users.SiteAdministrator = 1)
 or
 (Issues.ProjectId in [CurrentUserProjects]
  [AndIsusesPreFilter]
  and (exists(select top 1
                  Groups.UnrestrictedIssueAccess
              from UserGroups inner join Groups
                  on Groups.GroupCode = UserGroups.GroupCode
              where UserGroups.UserId = @CurrentUserId
              order by Groups.UnrestrictedIssueAccess desc)
        or
       (Issues.CreatedByUserId = @CurrentUserId
        or Issues.AssignedToUser = @CurrentUserId
        or exists(select *
                  from IssueHistory
                  where IssueHistory.IssueId = Issues.IssueId
                      and IssueHistory.OldAssignedToUser = @CurrentUserId)
       )
      )
 )
)
```

This macro can be used in a SQL query as follows:

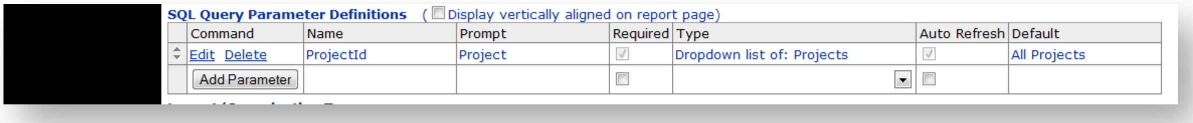
```
SELECT
    Issues.IssueId, Issues.Summary
FROM Issues
WHERE [IssueRowLevelSecurity]
    AND Issues.StatusCode = 'Open'
```

It is important to note that the [IssueRowLevelSecurity] macro only works with the Issues table (i.e. not an alias of that table). The macro also may not be suitable for all scenarios. SQL queries can be very complex and a simple macro cannot be guaranteed to always pass SQL's parsing rules.

It is recommended that issue row level security should always be used when querying the Issues table. This ensures that a report does not accidentally disclose information to users that they could not normally access.

It should also be noted that the RMTrack application provides **Field level security** – restricting which fields a user can see based on group membership. There is no macro provided to handle field level security. This should be controlled by what groups/users can access the report.

Parameterized Queries



[redacted]

SQL Query Parameter Definitions (□ Display vertically aligned on report page)

	Command	Name	Prompt	Required	Type	Auto Refresh	Default
⇕	Edit Delete	ProjectId	Project	☑	Dropdown list of: Projects	☑	All Projects
	Add Parameter			□	▾	□	

Often it is desirable to provide users of a report with parameters that control the results of a query. For example, allowing a user to select which project to report on.

When defining a report you can specify a list of parameters. These parameters are passed to the SQL query as actual SQL variables.

Any number of parameters can be added to a report. When the report is run the parameters can be presented *stacked* vertically, or across the top of the page by using the **Display vertically aligned** option.

To add a new parameter, fill in the required settings and click the **Add Parameter** button. Once a parameter has been added you can modify the settings by clicking the **Edit** link beside the parameter, making changes and the clicking **Update**.

The order of parameters can be adjusted by using the up and down arrows ⇕ at the far right of the parameter row.

A parameter consists of the following settings:

Setting	Meaning
Name	This is the SQL variable name (without the leading @ sign)
Prompt	This is the text that is shown to users when they run the report
Required	Indicates that the user must provide a non-blank value for the parameter in order for the report to be run.
Type	The type of parameter: checkbox, date*, daterange*, datetime*, number, text, or a dropdown list of one of the defined code tables (including Projects, Groups, Resolution Codes, Status Codes and Users).

Setting	Meaning
Auto Refresh	Indicates that whenever the parameter value has been changed the report is automatically regenerated. This setting is particularly useful for dropdowns and dates. If at least one parameter is not marked as “auto refresh” then a “Refresh” button will be displayed to the right of the parameters.
Default value	The default value for a parameter.

NOTE: Using date and date/time parameters in RMTrack reports can pose challenges due to local time zone settings. For example, if a user asks for “all records from yesterday” the right answer requires knowledge about the user’s preferred time zone. For assistance with queries involving dates and times, please contact support@rmtrack.com.

Parameter Example: A common use of a parameterized query is to allow a user to select a specific project or all projects they belong too. The parameter would be defined as:

- Parameter Name: **ProjectId**
- Prompt: **Project**
- Required: **Yes**
- Type: **Dropdown of Projects**
- Auto Refresh: **Yes**
- Default value: **All Projects (or -1)**

The query can then use the parameter as follows:

```
select
    Issues.AssignedtoUser,
    count(Issues.AssignedtoUser) as TotalIssues
from Issues
where (@ProjectId = -1 or Issues.ProjectId = @ProjectId)
group by Issues.AssignedtoUser
order by Issues.AssignedtoUser
```

NOTE: For a Project parameter a value of -1 indicates that All Projects was selected.

Report Layout/Organization Type

There are 7 different choices for the **Layout/organization** of reports:

Layout/Organization	Meaning
Report	A column oriented report where individual fields are displayed in columns. The layout can also include secondary rows, to a maximum of ten additional rows. Fields appearing on the additional rows are formatted with their caption to the left.
Report (auto format)	A column oriented report where individual fields are displayed in columns. The layout outputs all the columns from the result set using “General” formatting options. This layout type can be useful for simple reports or for reports where the columns of the result set are not pre-determined (eg. a result set generated by a crosstab)
BarChart/Simple	A bar chart of the data is displayed for a single series of data.
BarChart/Stacked	A bar chart of the data is displayed for multiple series of data, with each series “stacked” on top of each other (to a maximum of six series)
BarChart/Multi	A bar chart of the data is displayed for multiple series of data, with each series displayed side-by-side (to a maximum of six series)
LineChart/Simple	A line chart of the data is displayed for a single series of data.
LineChart/Multi	A line chart of the data is display for multiple series of data (to a maximum of six series).

Each layout/organization type has different parameters that are required to define the report.

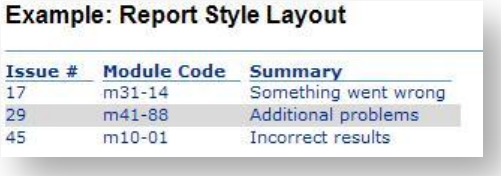
Report

This is the standard column and row report. You specify which columns from the query's result set are to be displayed and how the column is to be formatted. This style of report is very similar to the look of the My Issues page.

A typical result set for this type of report could look like:

IssueId	Module	Summary
17	m31-14	Something went wrong
29	m41-88	Additional problems
45	m10-01	Incorrect results

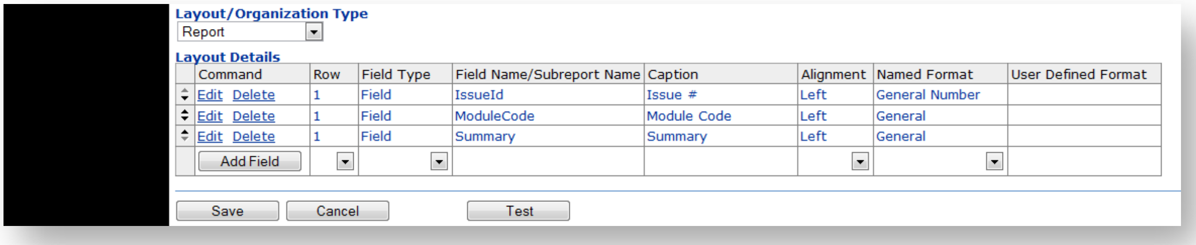
This data can produce a report like:



Example: Report Style Layout

Issue #	Module Code	Summary
17	m31-14	Something went wrong
29	m41-88	Additional problems
45	m10-01	Incorrect results

The layout settings for a **Report** Layout/Organization are as follows:



[redacted]

Layout/Organization Type

Report

Layout Details

	Command	Row	Field Type	Field Name/Subreport Name	Caption	Alignment	Named Format	User Defined Format
⬍	Edit Delete	1	Field	IssueId	Issue #	Left	General Number	
⬍	Edit Delete	1	Field	ModuleCode	Module Code	Left	General	
⬍	Edit Delete	1	Field	Summary	Summary	Left	General	
	Add Field							

Each row of the **Layout Details** represents a column in the report. Any number of columns can be added to a report.

To add a new column, fill in the required settings and click the **Add Field** button. Once a field has been added you can modify the settings by clicking the **Edit** link beside the field settings, making changes and the clicking **Update**.

The order of fields can be adjusted by using the up and down arrows ⬍ at the far right of the field row.

Setting	Meaning
Row	The row the field will appear on. The first row is formatted with column titles (the caption), the 2nd through Nth rows are formatted with side titles (the caption to the left of the field value).
Field Type	The following 4 types are supported: **Field:** A field from the query's result set **Hyperlink:** A clickable link to another report **Spacer:** A small gap of 25 pixels (used for alignment) **Subreport:** An entire subreport **NOTE: *Hyperlink* and *Sub-report* are explained in their own sections further on in this guide**
Field Name/ Subreport Name	If the Field Type is **Field** then this is the name of the field in the result set. If the Field Type is **Sub-Report** then this is the name of the sub-report
Caption	The column label for the field. Or the title for a sub-report.
Alignment	How the field is aligned: Left, Right or Center
Named Format	The format to use for the field. Values are supported: User Defined General General Date Short Date Short Datetime Medium Date Long Date Short Time Medium Time Long Time True/False Yes/No On/Off General Number Currency Fixed Standard Percent Scientific
User Defined Format	If the named format is **User Defined** then the field's format is based on the supplied custom format. Custom formats are produced using VBA Script standard functions (i.e. the same Format() function used by MS Word and MS Excel). Date formats are documented here: http://msdn.microsoft.com/en-us/library/aa263414(VS.60).aspx Numeric formats are defined here: http://msdn.microsoft.com/en-us/library/aa263416(VS.60).aspx Text formats are defined here: http://msdn.microsoft.com/en-us/library/aa263413(VS.60).aspx **NOTE: This setting has a different meaning when the field type is *Hyperlink*. Please see the section on Hyperlinks further on in this guide for an explanation.**

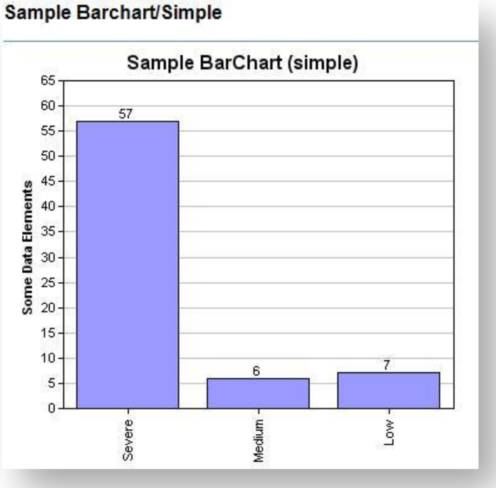
BarChart/Simple

The BarChart/Simple chart is the easiest of all the graphs. The query that produces the result set to chart must consist of at least two columns, the value to chart and the label/legend for the value. Additional columns are permitted, but are not used.

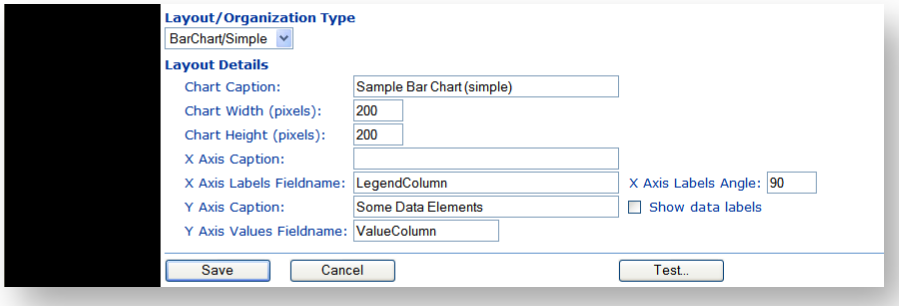
A typical result set looks like:

LegendColumn	ValueColumn
Severe	57
Low	7
Medium	6

This data can produce a chart like:



The layout settings for a **Barchart/Simple** Layout/Organization are as follows:



Layout/Organization Type

BarChart/Simple

Layout Details

Chart Caption:	Sample Bar Chart (simple)		
Chart Width (pixels):	200		
Chart Height (pixels):	200		
X Axis Caption:			
X Axis Labels Fieldname:	LegendColumn	X Axis Labels Angle:	90
Y Axis Caption:	Some Data Elements	☐ Show data labels	
Y Axis Values Fieldname:	ValueColumn		

The settings have the following meanings:

Setting	Meaning
Chart Caption	The title at the top of the chart.
Chart Width	The width of the chart in pixels.
Chart Height	The height of the chart in pixels.
X Axis Labels Fieldname	The name of the field in the query's result set that contains the text to be displayed as a label below each bar.
X Axis Labels Angle	The angle to display the labels below each bar. If the labels are too wide then placing them at an angle will make the chart more readable.
Y Axis Caption	The caption to be displayed beside the vertical axis.
Y Axis Values Fieldname	The name of the field in the query's result set that contains the values to be charted.

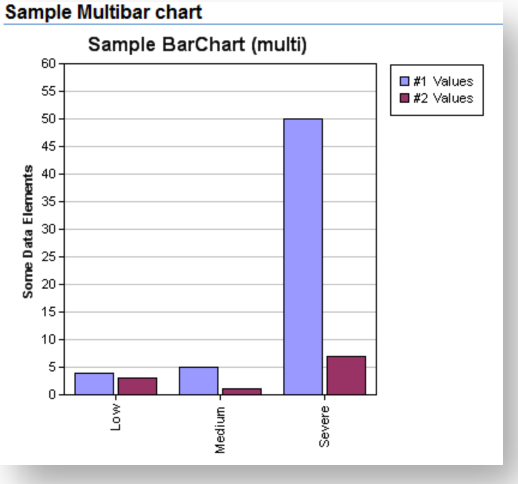
BarChart/Multi

The BarChart/Multi chart is similar to BarChart/Simple, except multiple sets of bars are charted. The query that produces the result set to chart must consist of at least three columns, the label/legend for the values, and at least two value fields.

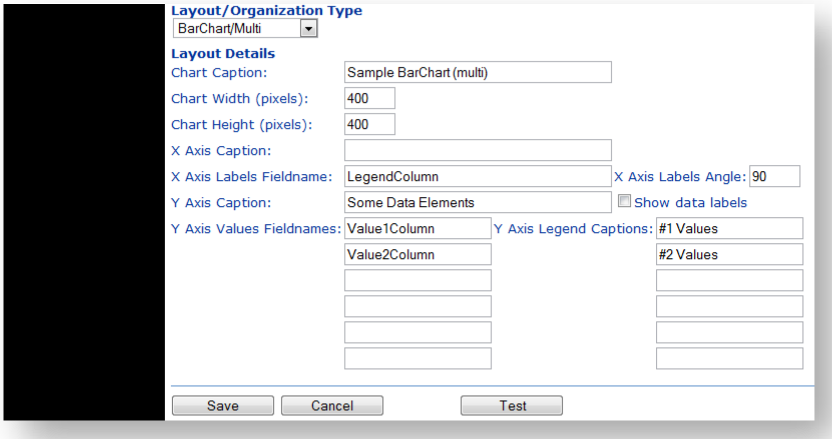
A typical result set looks like:

LegendColumn	Value1Column	Value2Column
Severe	50	7
Low	4	3
Medium	5	1

This data can produce a chart like:



The layout settings for a **BarChart/Multi** Layout/Organization are as follows:



Layout/Organization Type: BarChart/Multi

Layout Details

Chart Caption: Sample BarChart (multi)

Chart Width (pixels): 400

Chart Height (pixels): 400

X Axis Caption:

X Axis Labels Fieldname: LegendColumn X Axis Labels Angle: 90

Y Axis Caption: Some Data Elements □ Show data labels

Y Axis Values Fieldnames: Value1Column, Value2Column

Y Axis Legend Captions: #1 Values, #2 Values

The settings have the following meanings:

Setting	Meaning
Chart Caption	The title at the top of the chart.
Chart Width	The width of the chart in pixels.
Chart Height	The height of the chart in pixels.
X Axis Labels Fieldname	The name of the field in the query's result set that contains the text to be displayed as a label below each set of bars.
X Axis Labels Angle	The angle to display the labels below each bar. If the labels are too wide then placing them at an angle will make the chart more readable.
Y Axis Caption	The caption display beside the vertical axis.
Y Axis Values Fieldnames	A list of fieldnames in the query's resultset that contains the values to be charted. Up to 6 different fields can be charted together.
Y Axis Legend Captions	The text that appears in the legend box. Note that the legend box is always positioned at the top right.

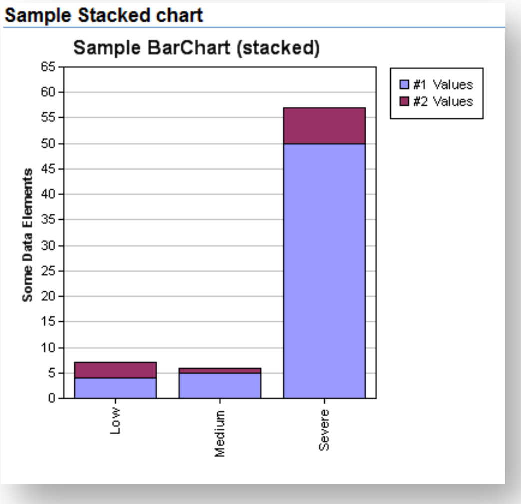
BarChart/Stacked

The BarChart/Stacked chart is the same as BarChart/Multi, except the multiple sets of bars are stacked instead of side-by-side. The query that produces the result set to chart must consist of at least three columns, the label/legend for the values, and at least two value fields.

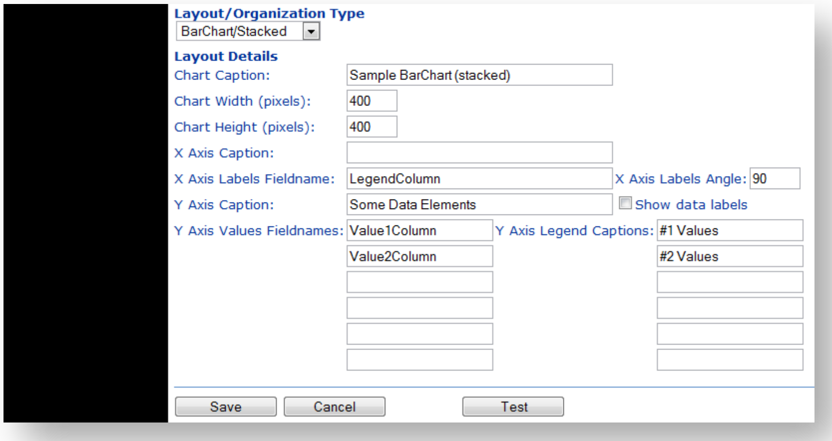
A typical result set looks like:

LegendColumn	Value1Column	Value2Column
Severe	50	7
Low	4	3
Medium	5	1

This data can produce a chart like:



The layout settings for a **BarChart/Stacked** Layout/Organization are as follows:



[redacted]

Layout/Organization Type

BarChart/Stacked

Layout Details

Chart Caption:	Sample BarChart (stacked)		
Chart Width (pixels):	400		
Chart Height (pixels):	400		
X Axis Caption:			
X Axis Labels Fieldname:	LegendColumn	X Axis Labels Angle:	90
Y Axis Caption:	Some Data Elements	□ Show data labels	
Y Axis Values Fieldnames:	Value1Column	Y Axis Legend Captions:	#1 Values
	Value2Column		#2 Values

The settings have the following meanings:

Setting	Meaning
Chart Caption	The title at the top of the chart.
Chart Width	The width of the chart in pixels.
Chart Height	The height of the chart in pixels.
X Axis Labels Fieldname	The name of the field in the query's result set that contains the text to be displayed as a label below each set of bars.
X Axis Labels Angle	The angle to display the labels below each bar. If the labels are too wide then placing them at an angle will make the chart more readable.
Y Axis Caption	The caption display beside the vertical axis.
Y Axis Values Fieldnames	A list of fieldnames in the query's result set that contains the values to be charted. Up to 6 different fields can be charted together.
Y Axis Legend Captions	The text that appears in the legend box. Note that the legend box is always positioned at the top right.

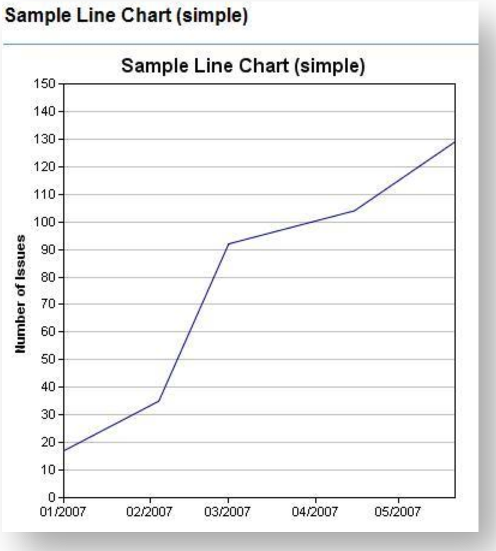
LineChart/Simple

The LineChart/Simple layout produces a line chart with a single line. The query that produces the result set to chart must consist of at least two columns, the x-axis values, and the y-axis values.

A typical result set looks like:

CreateDate	NumIssues
1-Jan-2007	17
4-Feb-2007	35
1-Mar-2007	92
15-Apr-2007	104
21-May-2007	129

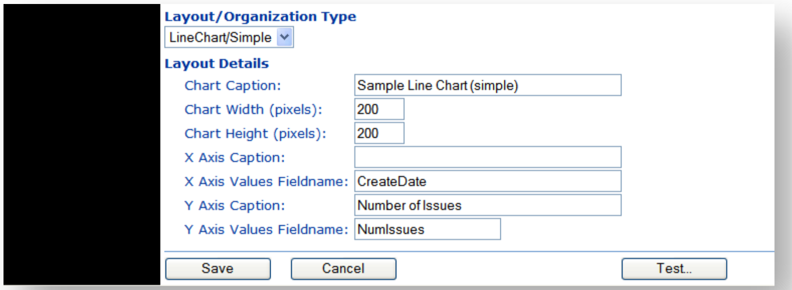
This data can produce a chart like:



The X-Axis values must be automatically convertible to a double precision floating point value. For this example the CreateDate column is a DATETIME type, which can be automatically converted to a double precision floating point value. Virtually any numeric/date value can be automatically converted. Character values, or strings, cannot be automatically converted and query you must do an explicit conversion using the SQL functions CAST or CONVERT.

The labels for the X axis ticks are automatically calculated.

The layout settings for a **LineChart/Simple** Layout/Organization are as follows:



[redacted]

Layout/Organization Type

LineChart/Simple

Layout Details

Chart Caption: Sample Line Chart (simple)

Chart Width (pixels): 200

Chart Height (pixels): 200

X Axis Caption:

X Axis Values Fieldname: CreateDate

Y Axis Caption: Number of Issues

Y Axis Values Fieldname: NumIssues

The settings have the following meanings:

Setting	Meaning
Chart Caption	The title at the top of the chart.
Chart Width	The width of the chart in pixels.
Chart Height	The height of the chart in pixels.
X Axis Values Fieldname	The name of the field in the query’s result set that contains the values for the X-Axis.
Y Axis Caption	The caption display beside the vertical axis.
Y Axis Values Fieldname	The name of the field in the query’s result set that contains the values to be charted.

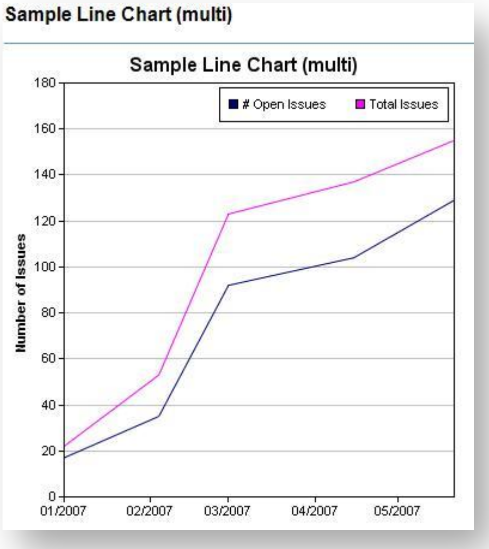
LineChart/Multi

The LineChart/Multi layout produces a line chart with two or more lines. The query that produces the result set to chart must consist of at least three columns, the x-axis values, and at least two columns of y-axis values.

A typical result set looks like:

CreateDate	NumOpenIssues	**TotalNumIssues**
1-Jan-2007	17	22
4-Feb-2007	35	53
1-Mar-2007	92	123
15-Apr-2007	104	137
21-May-2007	129	155

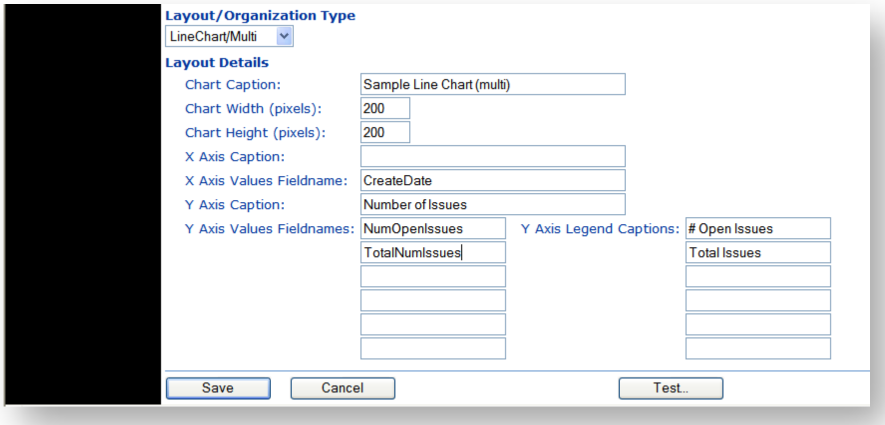
This data can produce a chart like:



The X-Axis values must be automatically convertible to a double precision floating point value. For this example the CreateDate column is a DATETIME type, which can be automatically converted to a double precision floating point value. Virtually any numeric/date value can be automatically converted. Character values, or strings, cannot be automatically converted and query should do an explicit conversion using the SQL functions CAST or CONVERT.

The labels for the X axis ticks are automatically calculated.

The layout settings for a **LineChart/Multi** Layout/Organization are as follows:



[redacted]

Layout/Organization Type

LineChart/Multi

Layout Details

Chart Caption: Sample Line Chart (multi)

Chart Width (pixels): 200

Chart Height (pixels): 200

X Axis Caption:

X Axis Values Fieldname: CreateDate

Y Axis Caption: Number of Issues

Y Axis Values Fieldnames: NumOpenIssues, TotalNumIssues

Y Axis Legend Captions: # Open Issues, Total Issues

The settings have the following meanings:

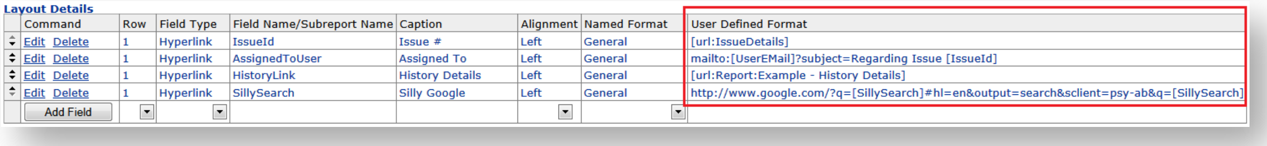
Setting	Meaning
Chart Caption	The title at the top of the chart.
Chart Width	The width of the chart in pixels.
Chart Height	The height of the chart in pixels.
X Axis Values Fieldname	The name of the field in the query's result set that contains the values for the X-Axis.
Y Axis Caption	The caption display beside the vertical axis.
Y Axis Values Fieldname	A list of fieldnames in the query's result set that contains the values to be charted. Up to 6 different fields can be charted together.
Y Axis Legend Captions	The text that appears in the legend box. Note that the legend box is always position at the top right.

Hyperlinks

When using a Report Layout/Organization type of **Report** you can have a **field type** of **hyperlink**. There are 4 types of hyperlinks available; a link to an issue details page, a link to an e-mail client (i.e. mailto:), a link to another RMTrack report, and a link to any other webpage.

When the report runs, these fields in the report behave as regular web page hyperlinks. They are underlined and can be clicked on to activate the link. It is recommended that the sample report **Example - Hyperlinks** be downloaded and installed to provide an example of hyperlink use.

The **User Defined Format** field controls the hyperlink's address (or more precisely the link's HREF attribute).



Layout Details

	Command	Row	Field Type	Field Name/Subreport Name	Caption	Alignment	Named Format	User Defined Format
⇕	Edit Delete	1	Hyperlink	IssueId	Issue #	Left	General	[url:IssueDetails]
⇕	Edit Delete	1	Hyperlink	AssignedToUser	Assigned To	Left	General	mailto:[UserEMail]?subject=Regarding Issue [IssueId]
⇕	Edit Delete	1	Hyperlink	HistoryLink	History Details	Left	General	[url:Report:Example - History Details]
⇕	Edit Delete	1	Hyperlink	SillySearch	Silly Google	Left	General	http://www.google.com/?q=[SillySearch]#hl=en&output=search&sclient=psy-ab&q=[SillySearch]
	Add Field	▾	▾			▾	▾	

Issue Details hyperlink

A macro is available to create a link in a report directly to the Issue Details for a particular issue. Generally this hyperlink is used on reports that display one row for each issue. The hyperlink would be placed on the IssueId (or IssueNumber) field but could also be on a summary or description field. The field **Issues.IssueId** must be part of the SQL query result set for the macro to work.

The user defined format for this type of hyperlink is:

[url:IssueDetails]

E-Mail (mailto:) hyperlink

An example of this hyperlink would be to place it on the AssignedToUser field or perhaps the CreatedByUserId field to facilitate sending an e-mail to these users. The **mailto:** hyperlink requires that the default e-mail client is known to the browser and this client will be opened when the hyperlink is selected. It is possible to pass result set fields in the mailto string, for example the user e-mail and the IssueId (both must be part of the SQL query result set

An example of a user defined format for this type of hyperlink is:

mailto:[UserEMail]?subject=Regarding Issue [IssueId]

Sub-Report hyperlink

A hyperlink in a report can be a link to another RMTrack Custom report. This allows for the creation of *drill-down* type reporting. For example you could have a report that sums all the issues assigned to each user and place a hyperlink on the user id that calls a another report that reports the details of each issue assigned to that user.

The user define format for this type of link is:

[url:Report:*SubReportName*]

(Where SubReportName is the name of the report to be called)

Webpage hyperlink

Any webpage URL can be addressed using this type of hyperlink. It is possible to pass result set fields in the URL query string. An example of its use might be referencing a knowledge base. For example, if you had a field on your issue that referenced a KB article number, you could pass that field in the hyperlink.

An example of a user defined format for this type of hyperlink is:

http://www.yourcompany.com/knowledgebase/?kb=[kbarticle]

(Where **kbarticle** is a field in the SQL query result set)

Sub-Reports

When using a Report Layout/Organization type of **Report** you can have a **field type** of **Subreport.**

Dashboard

One use for sub-reports is to provide a dash-board like interface of several unrelated reports.

To accomplish this, the main report should use a simple SQL query that returns a singe row. The layout can then include each sub-report as required. Each sub-report will only be displayed once, because the main query only returned a single row. Each sub-report’s query can use the same parameters as the main report and produces its own output. Please see the “Dashboard” report in the Sample Reports.

Parent/Child Sub-Reporting

Another use for sub-reports is to provide a parent-child style report. An example would be a report that contained one line for each project, and under each project it displays lines for each team member.

The main report (or parent, or top level) would be a query that returns a result set of one row for each project. The sub-report would be a query that returns a result set of rows for each project team member for the given project.

A sub-report is always executed once for each row in the parent report's query result set. If the parent report query did not return any rows then the sub-report will not be executed. If the main report query returned 2 rows, then the sub-report will be executed twice.

Parent/Child Sub-Report Example

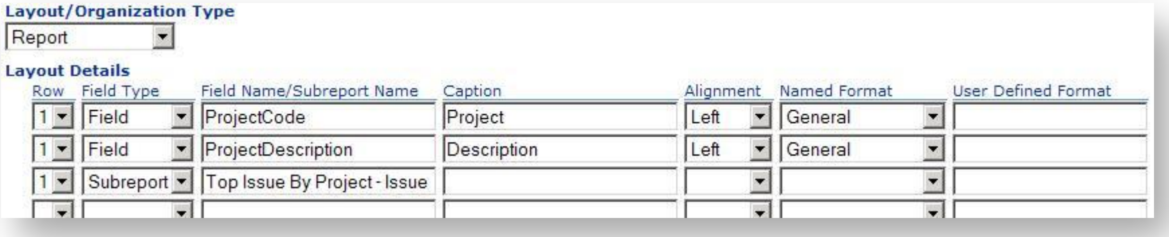
This example produces a report of projects that have at least one issue and the top 5 issues with the most history records. To create this report we'll need two report definitions:

- a parent report that selects all projects that have at least one issue
- a sub-report that that selects the top 5 issues with the most history records

The parent report's SQL query is straight-forward:

```
SELECT *
FROM Projects
WHERE EXISTS(SELECT * FROM Issues
             WHERE Issues.ProjectId = Projects.ProjectId)
ORDER BY ProjectCode
```

The parent report's layout is defined to show the project code, the project's description, and on a separate line the list of issues:



Layout/Organization Type

Report

Layout Details

Row	Field Type	Field Name/Subreport Name	Caption	Alignment	Named Format	User Defined Format
1	Field	ProjectCode	Project	Left	General	
1	Field	ProjectDescription	Description	Left	General	
1	Subreport	Top Issue By Project - Issue				

A key concept with the layout is the inclusion of the sub-report in parent report's layout (the third item listed in the Layout Details). The Field Name/Subreport Name field must contain the name of the sub-report itself. In this case we named the sub-report as **Top Issues by Projects - Issues sub-report**. The sub-report's definition is a bit more complicated. First of the sub-report should be marked as "Report is not published (hidden)" because we do not want the sub-report itself to appear in the left menu.

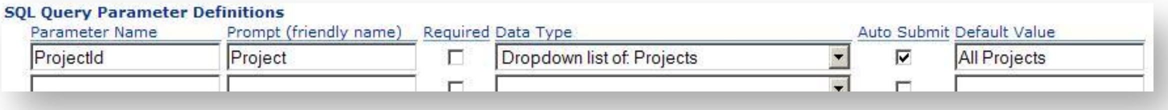
The SQL query look like:

```
SELECT TOP 5
    COUNT(IssueHistory.HistoryId) AS NumChanges,
    Issues.IssueId
FROM Issues INNER JOIN IssueHistory
    ON Issues.IssueId = IssueHistory.IssueId
WHERE Issues.ProjectId = @ProjectId
GROUP BY Issues.IssueId
ORDER BY COUNT(IssueHistory.HistoryId) DESC
```

Note that the query only returns the count of issue history records and the IssueId. Returning additional issue fields would require a more complex SQL due to the nature of the COUNT() and GROUP BY operators.

Also note the use of the "AS NumChanges" clause to provide a column name for the computation COUNT(IssueHistory.HistoryId). Columns that will appear in the report layout are required to have an explicit name, which SQL does not automatically define for computed values.

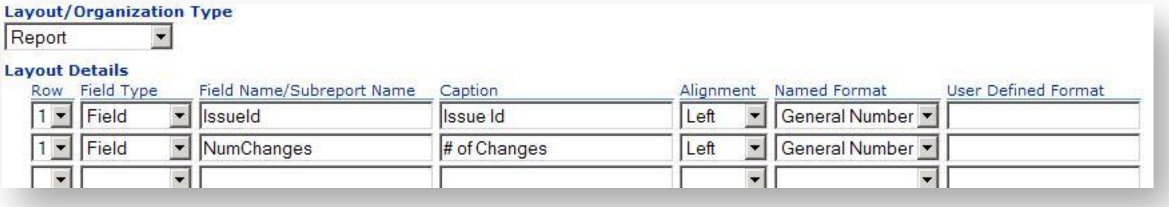
The query also uses a parameter, @ProjectId, that will be populated from the data returned by the parent report. This parameter must be formally defined in the list of parameters for the sub-report:



SQL Query Parameter Definitions

Parameter Name	Prompt (friendly name)	Required	Data Type	Auto Submit	Default Value
ProjectId	Project	□	Dropdown list of: Projects	☑	All Projects
		□		□	

The layout of the sub-report is just the issue id and the count of issues:



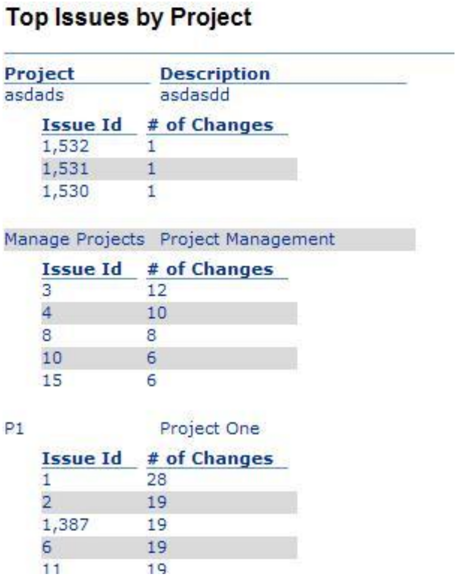
Layout/Organization Type

Report

Layout Details

Row	Field Type	Field Name/Subreport Name	Caption	Alignment	Named Format	User Defined Format
1	Field	IssueId	Issue Id	Left	General Number	
1	Field	NumChanges	# of Changes	Left	General Number	

That’s the full configuration of the parent report and sub-report. When accessed from the menu the report produces the following output:



Top Issues by Project

```
Project          Description
asdads           asdasdd

         Issue Id   # of Changes
         1,532      1
         1,531      1
         1,530      1

Manage Projects  Project Management

         Issue Id   # of Changes
         3          12
         4          10
         8          8
         10         6
         15         6

P1               Project One

         Issue Id   # of Changes
         1          28
         2          19
         1,387      19
         6          19
         11         19
```

Contact Information

At RMTrack we want your feedback! Please feel free to contact us, anytime, with questions, comments or inquiries:

General:	inquiries@RMTrack.com
Support:	support@RMTrack.com
Sales:	sales@RMTrack.com
Web-Site:	www.RMTrack.com
Phone:	416-929-1720
Toll Free:	1-888-225-2271 (Canada and United States)
Address:	RMTrack Issue Tracking Solutions Inc. 7 Walmer Road, Suite 1509 Toronto, Ontario Canada M5R 2W8

Issue Tracking for IT Professionals