

# ArchestrA™ Reports: Guidelines and Standards

**Invensys Systems, Inc.**

Technical Article Revision A

Last Revision: 2/12/08



## Copyright

© 2008 Invensys Systems, Inc. All Rights Reserved.

All rights reserved. No part of this documentation shall be reproduced, stored in a retrieval system, or transmitted by any means, electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of Invensys Systems, Inc. No copyright or patent liability is assumed with respect to the use of the information contained herein. Although every precaution has been taken in the preparation of this documentation, the publisher and the author assume no responsibility for errors or omissions. Neither is any liability assumed for damages resulting from the use of the information contained herein.

The information in this documentation is subject to change without notice and does not represent a commitment on the part of Invensys Systems, Inc. The software described in this documentation is furnished under a license or nondisclosure agreement. This software may be used or copied only in accordance with the terms of these agreements.

Invensys Systems, Inc.  
26561 Rancho Parkway South  
Lake Forest, CA 92630 U.S.A.  
(949) 727-3200

<http://www.wonderware.com>

For comments or suggestions about the product documentation, send an e-mail message to [productdocs@wonderware.com](mailto:productdocs@wonderware.com).

## Trademarks

All terms mentioned in this documentation that are known to be trademarks or service marks have been appropriately capitalized. Invensys Systems, Inc. cannot attest to the accuracy of this information. Use of a term in this documentation should not be regarded as affecting the validity of any trademark or service mark.

Alarm Logger, ActiveFactory, ArchestrA, Avantis, DBDump, DBLoad, DT Analyst, Factelligence, FactoryFocus, FactoryOffice, FactorySuite, FactorySuite A<sup>2</sup>, InBatch, InControl, IndustrialRAD, IndustrialSQL Server, InTouch, MaintenanceSuite, MuniSuite, QI Analyst, SCADAAlarm, SCADASuite, SuiteLink, SuiteVoyager, WindowMaker, WindowViewer, Wonderware, Wonderware Factelligence, and Wonderware Logger are trademarks of Invensys plc, its subsidiaries and affiliates. All other brands may be trademarks of their respective owners.

# Contents

Documentation Conventions.....	7
Technical Support .....	7
<b>Chapter 1 Creating ArcestraA Reports .....</b>	<b>9</b>
Glossary of Terms.....	9
References .....	10
Overview .....	10
Document Overview .....	10
Creating a New Report .....	10
Using Metric Units.....	11
Creating New Visual Studio and Report Server Projects.....	12
Modifying the Report Template.....	16
Working with Report Sections .....	16
Associating Report Sections with Datasets .....	16
Creating SQL Objects and Report Datasets .....	17
Creating a Linked Server.....	17
Defining the Required Views .....	17
Creating a Dataset .....	18
Associating the Report Section to Your Dataset.....	19
Working with Report Labels.....	20
Modifying the Label Dataset .....	22
Using Report Filter Parameters .....	23
Removing Unused Report Parameters.....	24
Displaying User Selections for Report Filtering.....	25

Controlling Report Section Display.....	25
Testing and Preparing for Deployment.....	26
Tips and Guidelines .....	28
Chart Rendering.....	28
Limit Outlier Values .....	28
Horizontal Bar Charts .....	29
Data in Charts is Not Exported.....	29
Configuring the Report Layout.....	30
Using Rectangles as Report Sections .....	30
Check the Print Layout.....	30
Text Box Length .....	31
Report Colors .....	31
Report Contents.....	32
Report Structure.....	32
Hyperlinks .....	32
Report Parameters .....	33
Date/Time Parameters with Time Picker .....	33
Multi-valued Parameters: (Select All).....	33
Long List of Values Replacement .....	34
Query-based Default Values .....	34
Floating Point Default Values .....	35
Preparing for Localization .....	35
Number and date/time format.....	35
Currency Format .....	35
Report Language .....	35
Configuring SQL Objects .....	36
Object Naming.....	36
UTC and Time Zone Handling .....	36
Displaying Times in Reports .....	37
Date/Time Data .....	37
Date/time Formatting .....	37
Data In UTC .....	37
Data In Server Time .....	38
Decode Values.....	39
Schema Usage .....	39
Naming Conventions.....	39
Report Templates Description .....	40
Colors .....	42
Page Size.....	43
Page Header .....	43
Page Footer .....	43
Reports Sections .....	43
Report Section Headings.....	44

- Report Definition Summary Section ..... 44
- Tabular Sections..... 44
- Matrix Sections..... 45
- List Sections..... 45
- Chart Sections ..... 45
- Units of Distance ..... 45
- Testing Your Reports ..... 46
  - Verify Data Returned by SQL Objects ..... 46
  - Check Calculations..... 46
  - Check Report Parameters ..... 46
    - Query Based Parameter Contents..... 46
    - Parameter Dependency ..... 47
    - Data filtering parameters ..... 47
    - Parameter Behavior ..... 47
  - Check Drill-down Paths ..... 47
  - Check Report Localization ..... 47
  - Check Printed Report Format ..... 48
  - Check Expand/Collapse Behavior..... 48
  - Check Color Application..... 49
  - Check Chart Limits ..... 49
  - Check Time Zone Selection ..... 49
- Index ..... 51**



---

# Welcome

You can view this document online or you can print it, in part or whole, by using the print feature in Adobe Acrobat Reader.

This guide assumes you know how to use Microsoft Windows, including navigating menus, moving from application to application, and moving objects on the screen. If you need help with these tasks, see the Microsoft online help.

## Documentation Conventions

This documentation uses the following conventions:

Convention	Used for
Initial Capitals	Paths and file names.
<b>Bold</b>	Menus, commands, dialog box names, and dialog box options.
Monospace	Code samples and display text.

## Technical Support

Wonderware Technical Support offers a variety of support options to answer any questions on Wonderware products and their implementation.

Before you contact Technical Support, refer to the relevant section(s) in this documentation for a possible solution to the problem. If you need to contact technical support for help, have the following information ready:

- The type and version of the operating system you are using. For example, Microsoft Windows XP, SP1.

- Details of how to recreate the problem.
- The exact wording of the error messages you saw.
- Any relevant output listing from the Log Viewer or any other diagnostic applications.
- Details of what you did to try to solve the problem(s) and your results.
- If known, the Wonderware Technical Support case number assigned to your problem, if this is an ongoing problem.



---

# Chapter 2

## Creating ArcestrA Reports

This document provides information that can be used by developers building ArcestrA Reports for Wonderware Information Server using Microsoft SQL Server 2005 Reporting Services.

This document does not replace the Administration Guide provided with Wonderware Information Server (WIS) but rather supplements it by providing:

- Detailed information on the standards used to create the report templates provided with Wonderware Information Server.
- Tips and guidelines on how to create new ArcestrA reports.

### Glossary of Terms

The following terms are used in this document:

---

<b>Term</b>	<b>Definition</b>
BIDS	SQL Server <u>B</u> usiness <u>I</u> ntelligence <u>D</u> evelopment <u>S</u> tudio
RDL	<u>R</u> eport <u>D</u> efinition <u>L</u> anguage (XML document defining the report in RS)
RS	Microsoft SQL Server <u>R</u> eporting <u>S</u> ervices
WIS	<u>W</u> onderware <u>I</u> nformation <u>S</u> erver

---

## References

*Wonderware Information Server Administration Guide*, Chapter 19.

## Overview

Wonderware Information Server (WIS) 3.0 introduced ArcestrA Reports in order to extend its information display capabilities. ArcestrA Reports are based on Microsoft SQL Server 2005 Reporting Services (RS).

ArcestrA Reports provide report templates and additional functions that make it easier to create reports for the Industrial world. In addition, WIS provides user-friendly controls that facilitate parameter input when tag names and date ranges are used.

WIS also provides a report deployment utility that provides report deployment to multiple servers, for a multi-site enterprise.

## Document Overview

This document provides information on the standards used to create the report templates and provides tips and guidelines on how to use these templates to create new reports and on how to create the SQL objects that support these reports.

The first section of this document provides a tutorial on how to create a report using one of the report templates as a starting point.

The next section provides tips and guidelines on how to build and structure an ArcestrA report. A description of the report templates follows, with details on the fonts, colors and functions.

Finally, the document provides a check list that can be used to verify and test the resulting ArcestrA Report.

## Creating a New Report

This section provides a sample step-by-step procedure on how to create an ArcestrA report based on one of the 2 templates provided with WIS.

---

**Note** It is strongly suggested you read Chapter 19 of the *Wonderware Information Server Administration Guide* prior to reading this section. Training on Microsoft SQL Server Reporting Services is also recommended.

---

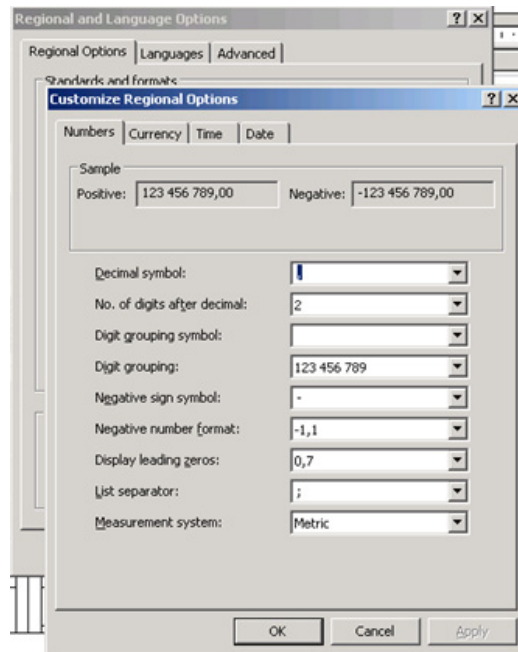
This procedure assumes that Wonderware Information Server is installed with the ArcestrA Reports option. The procedure also assumes that SQL Server Business Intelligence Development Studio (BIDS) is also installed on the WIS server and that development of the reports is done from that instance of BIDS.

## Using Metric Units

Report templates were made using metric units, which makes it easier to control object sizing and positioning. It is therefore highly recommended to switch to metric units when working with the BIDS.

### To switch to metric units

- 1 Open the Control Panel.
- 2 Click **Regional and Language Options**, then the **Regional Options** tab.
- 3 Click the **Customize** button.
- 4 Click the **Numbers** tab.
- 5 Select **Metric** from the **Measurement system** list.



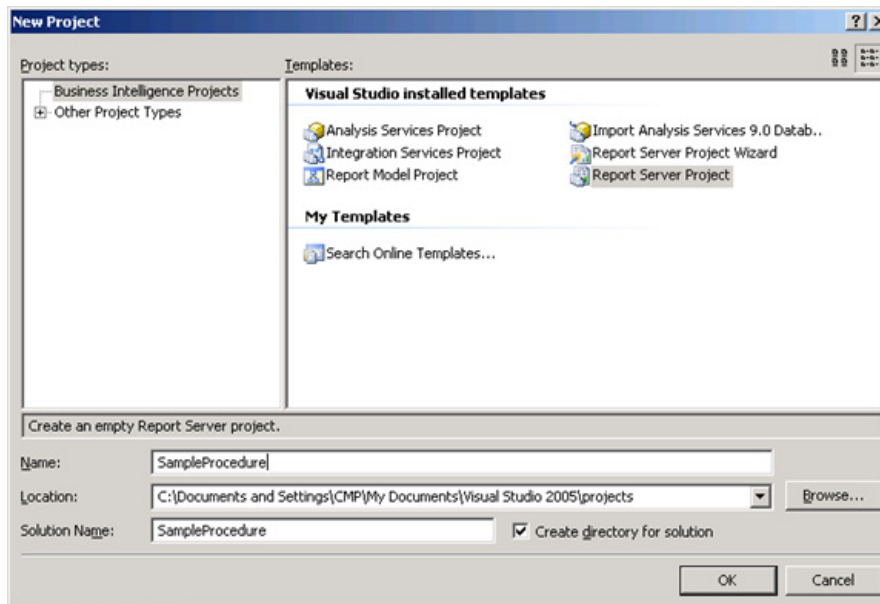
- 6 Click **OK**, then **OK** to exit the Regional and Language options.

## Creating New Visual Studio and Report Server Projects

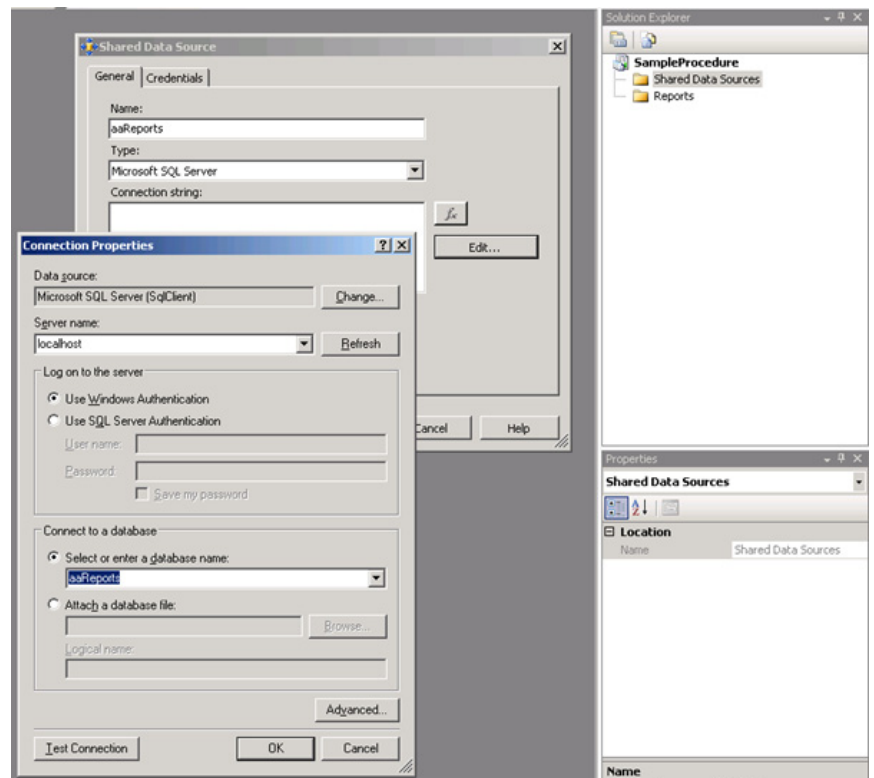
Next, you must create a project in Visual Studio.

### To create a new project

- 1 Start BIDS on the WIS Server node and create a new Report Server Project.

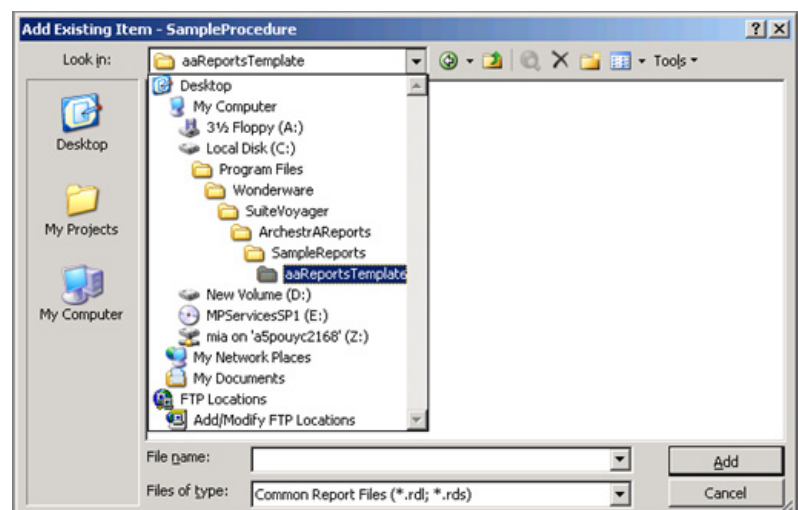


## 2 Create a new Data Source in the Solution Explorer.

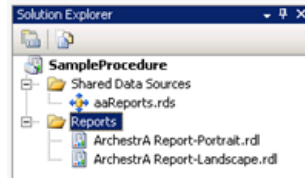


## 3 Add the 2 ArchestrA Reports templates to the Reports folder in Solution Explorer (Add existing item).

The location of these files is described in Chapter 19 of the WIS Administration Guide. They are typically found in the **Program Files** folder, or on the WIS installation CD.

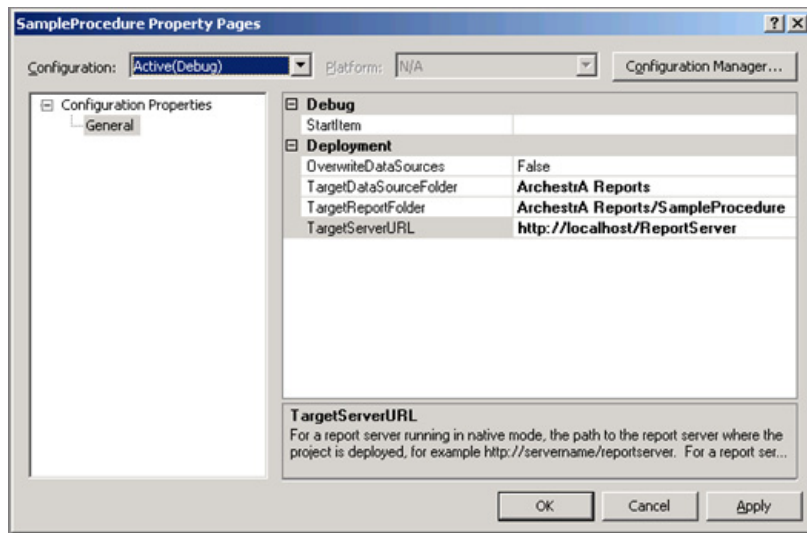


The following graphic shows the two report templates:



- 4 Modify the Project Properties for Arcestra Reports as shown in the following graphic.

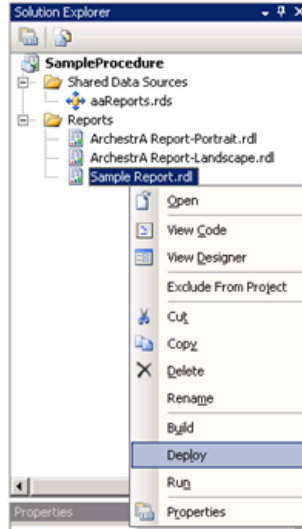
**Note** You should modify the TargetReportFolder with a name representing your project: Arcestra Reports/YourProjectName.



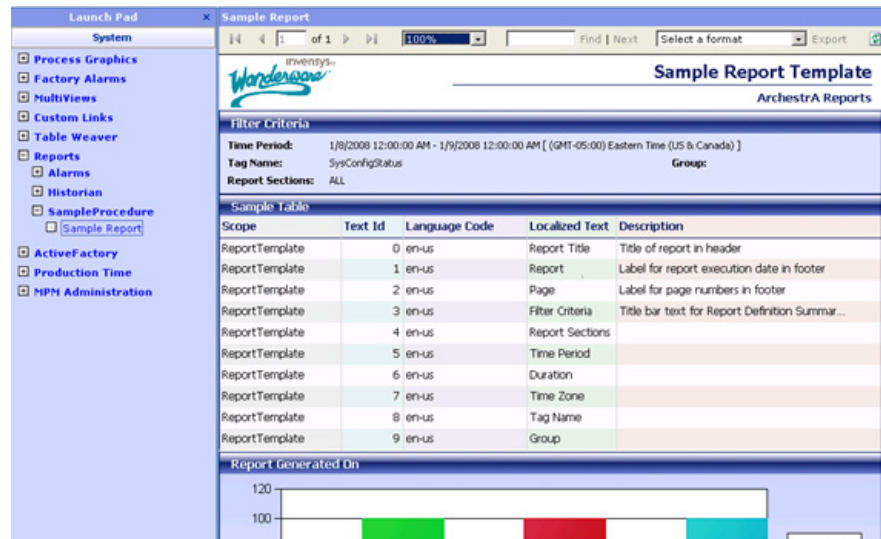
When you deploy the reports, they automatically appear in WIS, under the **Reports** node.

- 5 Make a copy of the desired template depending on the format of the report you want to create (portrait or landscape).
- 6 Select the report template and from the BIDS menu, use Copy/Paste. You can rename the report with a meaningful name.

- 7 Deploy your report. Report templates are fully functional examples. Therefore, you can already test your installation by deploying your report and testing it in WIS. The report displays data taken from the aaReports database installed with Arcestra Reports.



An example of a deployed report is shown below:



## Modifying the Report Template

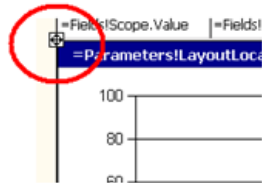
You are now ready to modify the template for your report content.

Open the report in layout mode by double-clicking on it in the Solution Explorer. The report contains an example of a table, a chart and a matrix section.

### Working with Report Sections

Depending on your report layout, you can duplicate the sections that you need and eliminate the ones that you do not need.

Each section is a rectangle, which facilitates the handling of the section in terms of sizing and movement. You only need to select the rectangle, identified by the cross-arrow in the upper-left corner of the section when you move the mouse over the section, as shown in the following graphic:



By clicking the sizing icon, you select the entire rectangle and its content. You can now move, duplicate or move that rectangle as needed.

Note that it is important to respect the width of the report. The template is set at its maximum limit to fit A4 and Letter paper sizes: 18.95 cm for portrait and 26.9 cm for landscape.

### Associating Report Sections with Datasets

Each of the report sections must be associated to a Dataset. Datasets are queries executed against the aaReports database. In order to access data in other databases, views referring to linked servers are created in the aaReports database.

You can use stored procedures running against views defined in the aaReports database for more complex queries. These stored procedures are also defined in the aaReports database. Report Datasets then refer directly to the views or stored procedures defined in the aaReports database.

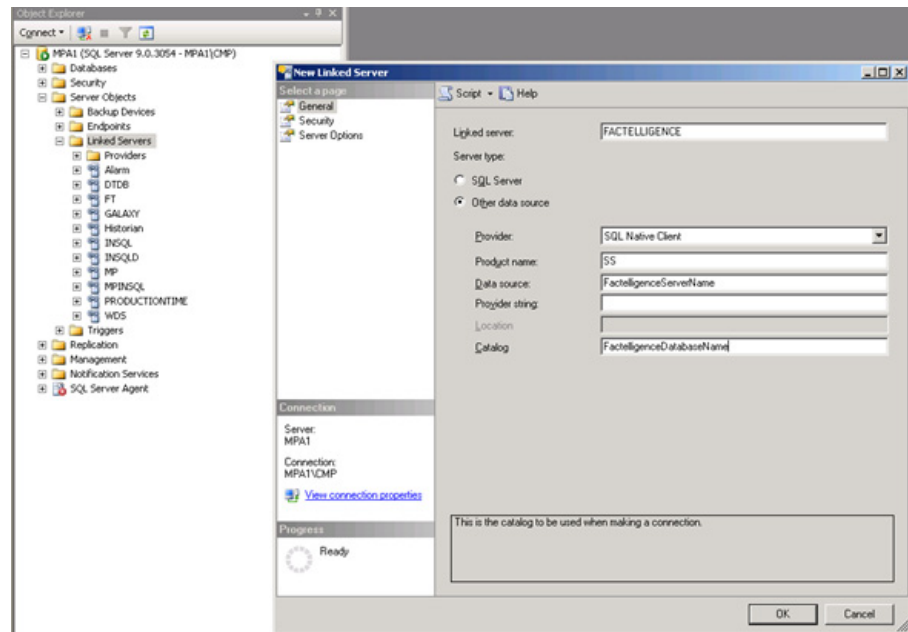


## Creating SQL Objects and Report Datasets

The following steps explain how to create the supporting SQL objects and then the report Datasets. We will use a Factelligence remote database as an example.

### Creating a Linked Server

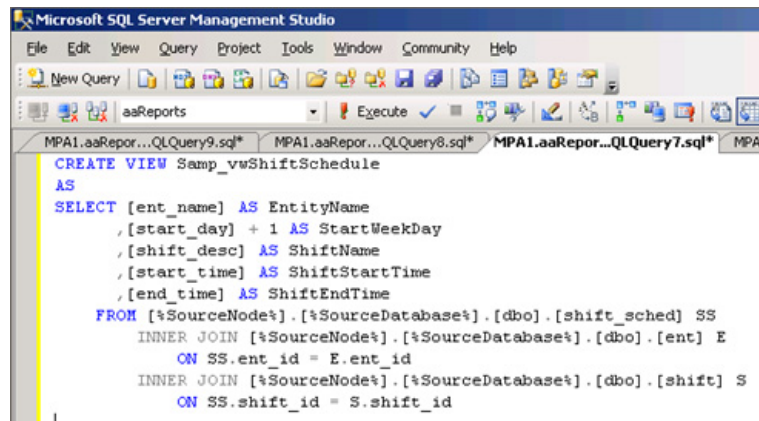
Create a linked server to access the remote data. This can be done in SQL Server Management Studio. You may need to review the security settings of the linked server, depending on your network structure.



### Defining the Required Views

You then need to define the view(s) required to access the data. The view is created in the aaReports database.

Using SQL Server Management Studio, here is an example:



The [%SourceNode%] and [%SourceDatabase%] values are placeholders that can be replaced by the remote server and database names, respectively.

It is a good practice to add a prefix to the view name that uniquely identifies the project you are working on. The aaReports database may be the host for several other report sets and their associated SQL objects. The prefix helps identify which objects are associated with which reports. In the example above **Samp\_** was used as a prefix.

## Creating a Dataset

You can now create a Dataset that refers to the view.

---

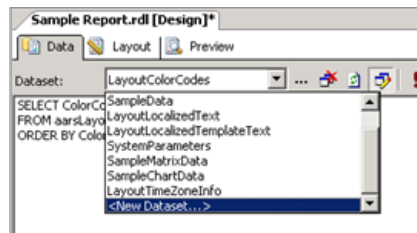
**Note** The procedure is similar for a stored procedure.

---

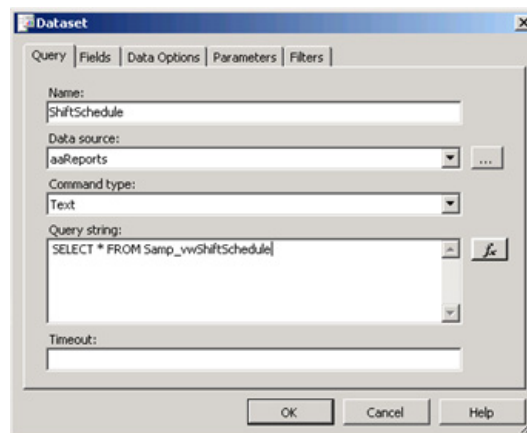
### To create a Dataset for the view

- 1 In our example, the report Dataset statement is:  

```
SELECT * FROM Samp_vwShiftSchedule
```
- 2 Click the **Data** tab in BIDS, as shown below.



- 3 Select **New Dataset** from the list, and add the query string along with other information.



- 4 Click **OK**.
- 5 Run the query to validate the Dataset by clicking the red exclamation mark in the toolbar.

## Associating the Report Section to Your Dataset

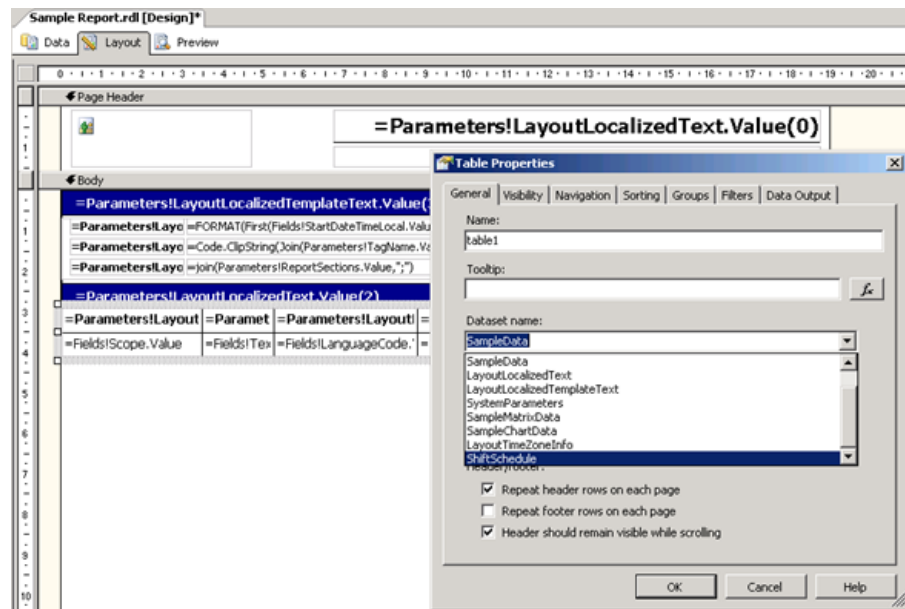
Next, you need to associate your report section to your newly created Dataset.

In this example, we will use a Table section to display the query results.

The following steps explain how to modify the table in the report template to point to the new Dataset.

### To associate the table to your Dataset

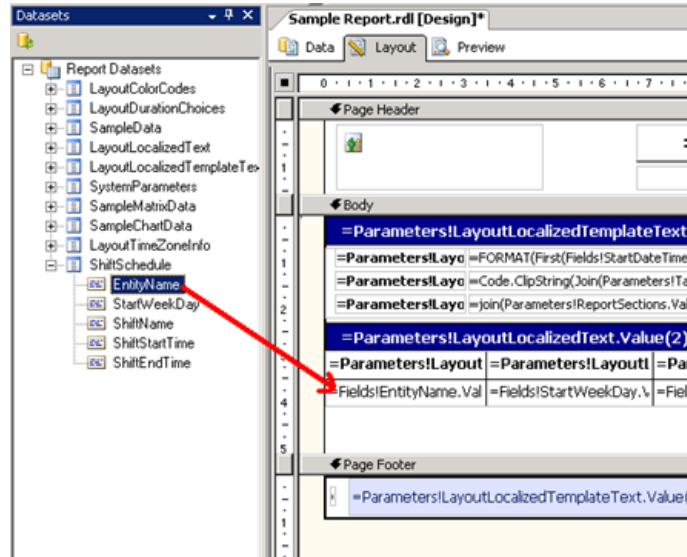
- 1 Click the **Layout** tab in BIDS and select the table you want to associate to the Dataset.
- 2 Right-click the table and select **Properties**.
- 3 Change the Dataset name to your newly created Dataset.



You're ready to associate the Dataset fields to each of the table columns.

### To associate Dataset fields to table columns

- 1 Drag-and-drop the fields from the Dataset folder to the appropriate table column. If you do not see the Dataset panel on the left hand pane, you can restore it by selecting View/Datasets from the main BIDS menu.



The template displays a tooltip expression for each of the columns in the table.

- 2 Remove the expression or change it to a valid field in the Dataset. Otherwise, you will get an error at build time stating that the tooltip expression refers to a field that does not exist.

### Working with Report Labels

All labels in the report templates (including the column headers in the table) are formulas, such as the following:

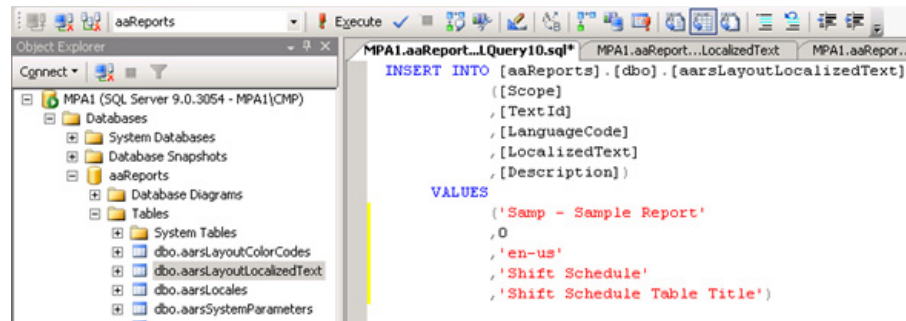
```
=Parameters!LayoutLocalizedText.Value(4)
```

These formulas allow you to create a report that can be displayed in multiple languages since the labels are extracted from a table located in the aaReports database.

In order to take advantage of this functionality, you need to add records in the aaRSLayoutLocalizedText table in the aaReports database.

The insertion of these records will need to be added to the deployment package. It is a good idea to create a file containing your INSERT statements that can then be appended to the deployment package later on.

The following is an example of an INSERT statement syntax:



An INSERT statement is required for each label in the report.

The Scope field uniquely identifies the report. It is a good practice here to prefix the report name with the same prefix used for SQL objects ("Samp" in our example).

The TextID field is incremented for each new label, starting from 0 and incrementing by 1.

---

**Caution** Do not skip a TextID number, because labels are accessed by row number, not by the TextID itself.

---

- The LanguageCode field is the locale ID for the language.
- The LocalizedText field is the actual label text displayed in the report.
- The Description field describes the label.

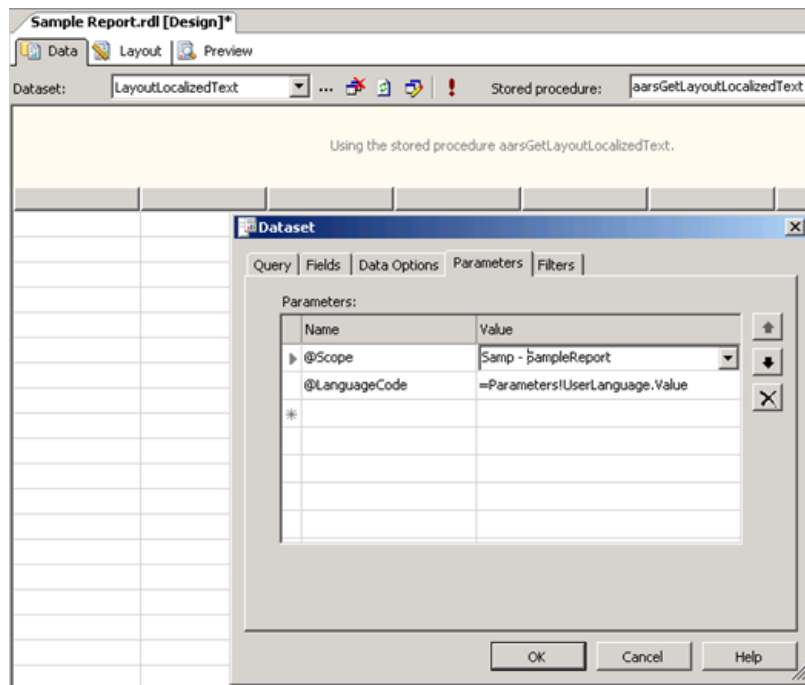
Once all labels have been entered for the default language, you can duplicate the INSERT statements, and replace the LanguageCode field and LocalizedText fields for another language. You repeat this process for all supported languages.

In the report, all label fields should use the formula shown above. The index of **Values** corresponds to the desired label. In the example above, the label associated with **TextId = 4** will be used.

## Modifying the Label Dataset

Once all labels have been defined, you must modify the report Dataset (LayoutLocalizedText) extracting these labels from the aaReports database.

The Dataset uses a stored procedure to extract the labels in the language of the client browser. The **@Scope** parameter must be changed to the Scope that was used in the previous step.

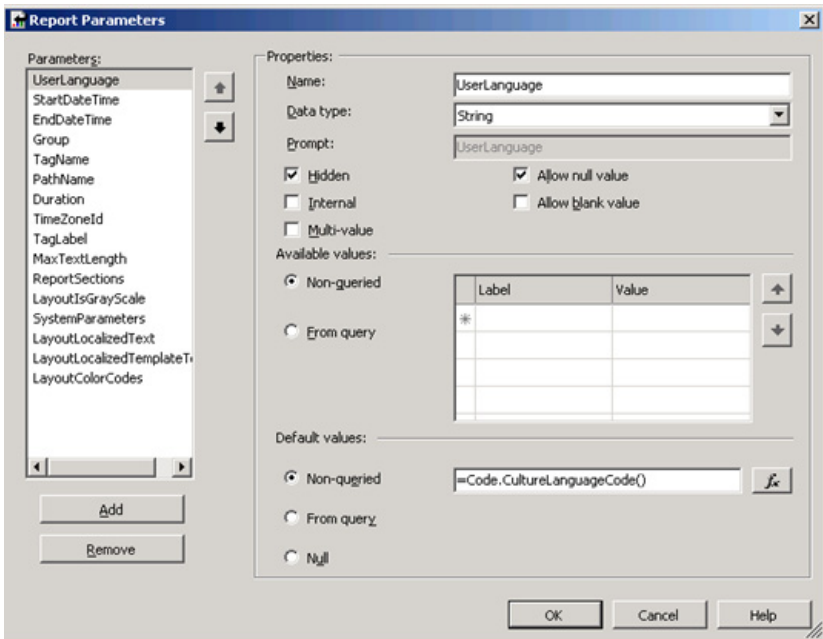


You can repeat the procedures for all reports sections:

- Creating a Dataset
- Associating it to the report section; and
- Defining report labels.

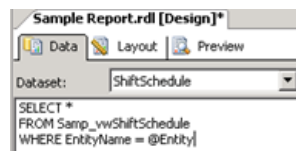
## Using Report Filter Parameters

Report parameters may be required to filter Datasets. Common report parameters are already included in the report template. Report parameters are accessed from the BIDS menu: Report, Report Parameters.



Parameters can be added as needed. Parameters for stored procedures are automatically added when a new stored procedure is referenced by a Dataset.

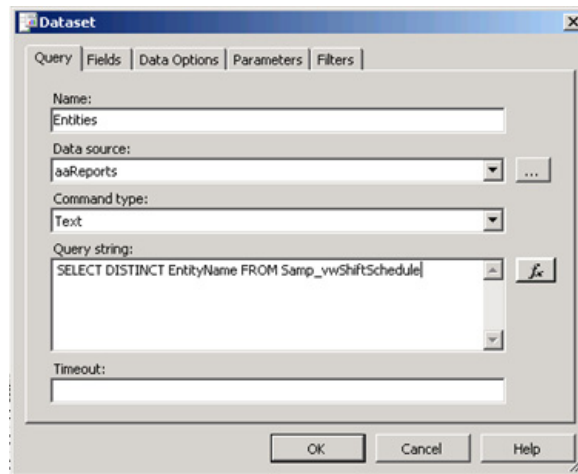
A parameter for a text query can be easily added by adding a WHERE clause and referring to a parameter prefixed with a "@" symbol. For example:



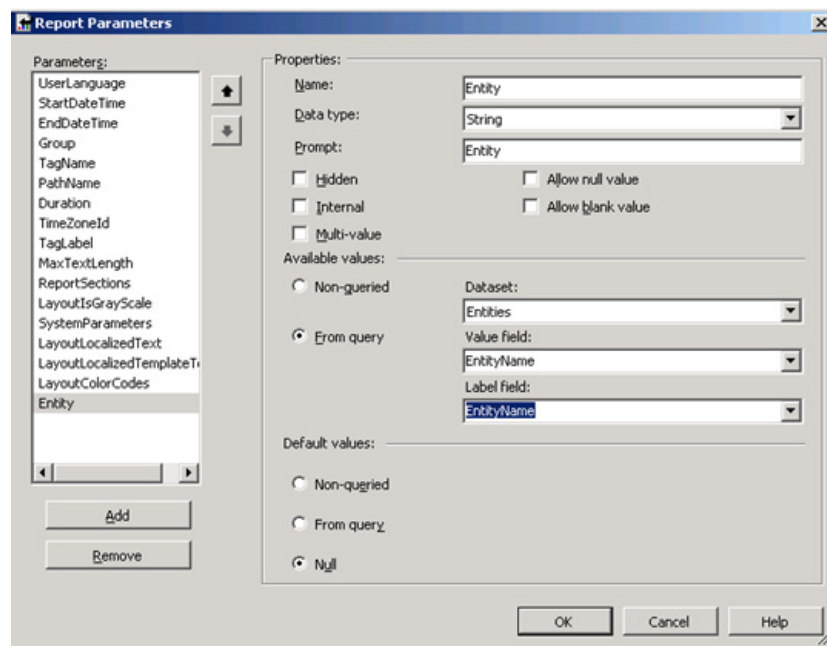
This query automatically adds a parameter named **Entity**. When you execute the report, you will be prompted to enter a value for the Entity parameter.

If you want to display a list of valid values for a given parameter so that the user can select the value from a list, define the list of available values using a query.

In the following graphic, we add a Dataset called **Entities**:



Then we modify the report parameter so that its set of available values comes from the Dataset Entities:



### Removing Unused Report Parameters

Unused report parameters associated to the Tag Picker and/or to the Time Picker can be removed if these controls are not used. For example, if you are not planning to use the Tag Picker control, you can remove the TagName, PathName and Group parameters.



If you are not planning to use the Time Picker control, you can remove the `StartDateTime`, `EndDateTime`, `Duration` and `TimeZoneId` parameters. The presence of those parameters determine whether the controls are displayed in the WIS parameter page.

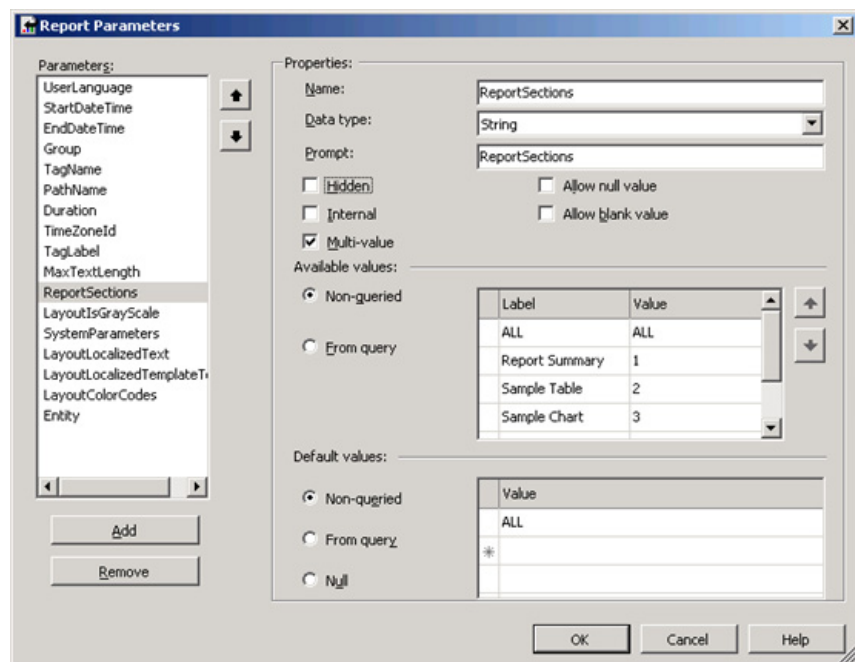
## Displaying User Selections for Report Filtering

It is a good practice to include a section in your report that displays available selections for data filtering report parameters. The report templates already include such a section (called **Filter Criteria** by default) that can be adapted for your report parameters.

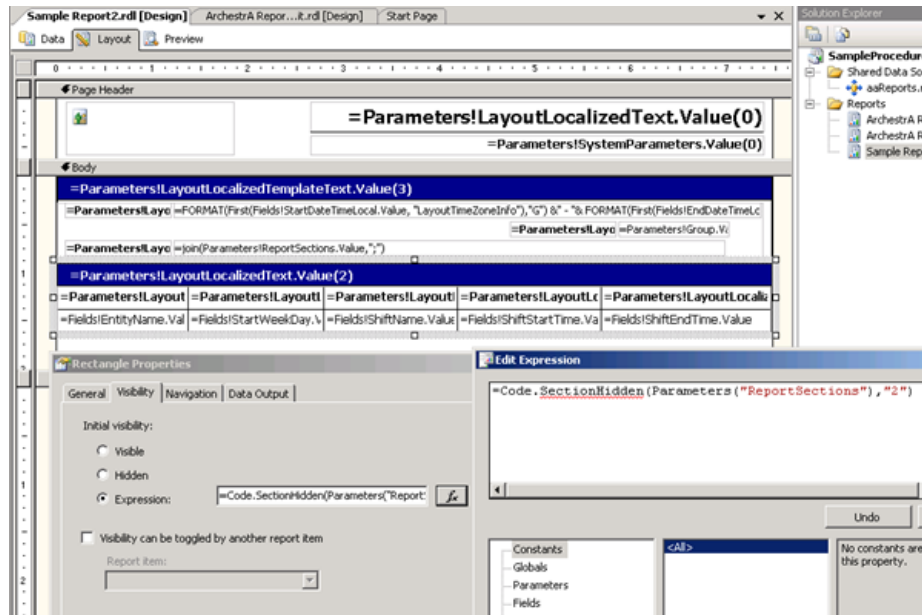
This section in the report template includes the parameter selection for the parameters related to the Tag and Time Picker controls. You can add and remove any parameters. In general, only those parameters that filter data in the report are displayed.

## Controlling Report Section Display

The report template introduces another mechanism that allows report consumers to display only the report sections that they want at execution time. This is controlled by the **ReportSections** parameter. Each report section is given a label and a value, which are entered in the `ReportSections` parameter:



Each report section is encapsulated in a rectangle object, so that the visibility attribute of the rectangle controls whether the section is displayed or not. The expression used in the visibility attribute of the rectangle is as follows:



## Testing and Preparing for Deployment

At this point, the report should be functional. It can then be built, deployed and tested. A testing checklist is provided later in this document.

Finally, once the report is considered complete, a deployment package for WIS can be prepared. The Report Deployment utility provided with WIS makes deployment of reports very easy and efficient.

The Administration Guide provides more information on creating. A deployment package includes:

- The RDL files (the report definition files created by BIDS).
- A SQL script that creates all database objects required to support the report.
- A configuration file (XML document).

Here is an example of a typical SQL script:

```

use [%ReportsDatabase%]
SET ANSI_NULLS ON
SET QUOTED_IDENTIFIER ON
GO

-- Drop all objects related to these reports, if they exist.
IF EXISTS (SELECT * FROM sys.views WHERE object_id = OBJECT_ID(N'[%dbo].[Samp_vwShiftSchedule]'))
    DROP VIEW [%dbo].[Samp_vwShiftSchedule]
GO

-- Create database objects supporting the reports.
CREATE VIEW Samp_vwShiftSchedule
AS
SELECT [ent_name] AS EntityName
      ,[start_day] + 1 AS StartWeekDay
      ,[shift_desc] AS ShiftName
      ,[start_time] AS ShiftStartTime
      ,[end_time] AS ShiftEndTime
FROM [%SourceNode%].[%SourceDatabase%].[%dbo].[shift_sched] SS
  INNER JOIN [%SourceNode%].[%SourceDatabase%].[%dbo].[ent] E
    ON SS.ent_id = E.ent_id
  INNER JOIN [%SourceNode%].[%SourceDatabase%].[%dbo].[shift] S
    ON SS.shift_id = S.shift_id
|
-- Create all localized text for the report, in all supported languages
DELETE FROM [aarsLayoutLocalizedText] WHERE [Scope] LIKE 'Samp -%'

INSERT INTO [aarsLayoutLocalizedText]
VALUES ('Samp - Sample Report',0,'en-us','Sample Report','Sample Report Title')
INSERT INTO [aarsLayoutLocalizedText]
VALUES ('Samp - Sample Report',1,'en-us','N/A','Available')

```

Here is an example configuration file:

```

<?xml version="1.0" encoding="utf-8"?>
<ReportSources xmlns = "aaReportsSchema.xsd">
  <SourceDatabase name="aaReports">
    <WebConfig>
      <appSetting>
      </appSetting>
    </WebConfig>
    <ReportServer>
      <Deploy file="aaReports.sql" />
    </ReportServer>
    <Reporting>
      <Folder name="Sample Reports" description="Sample Reports">
        <Report name="Sample Report 1" file="SampleReport1.rdl" />
        <Report name="Sample Report 2" file="SampleReport2.rdl" />
      </Folder>
    </Reporting>
  </SourceDatabase>
</ReportSources>

```

## Tips and Guidelines

The following section provides tips and guidelines for developing Arcestra Reports.

### Chart Rendering

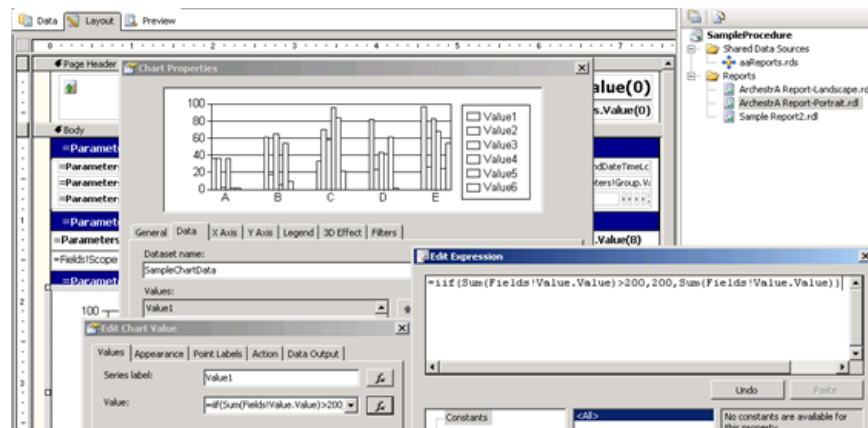
Charts and trends in RS 2005 are either scaled automatically based on the spread of the underlying data, or you can fix the scale at design time. Unfortunately, scales cannot be based on expressions at this time.

#### Limit Outlier Values

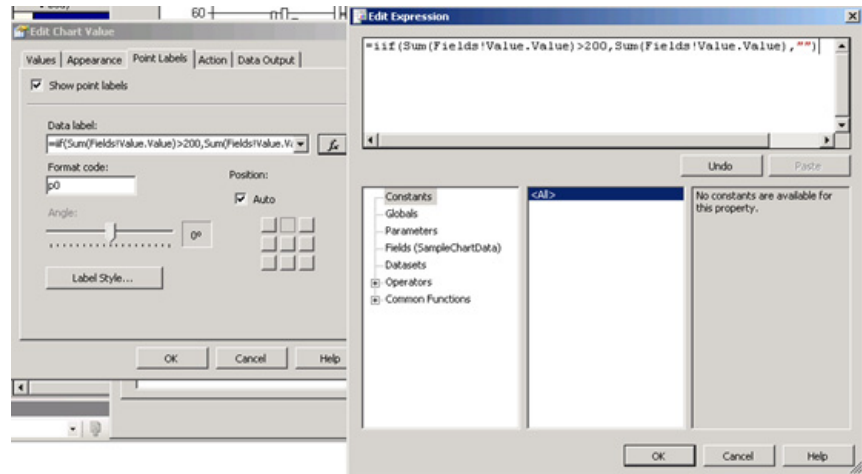
When using automatic chart scaling, outlier values in the data can offset the scale of the chart to a point where it becomes unreadable. Options to avoid this situation are:

- Fix the scale to a known range and limit values to a minimum and/or maximum.
- Add a point label with the actual outlier value.

Here is an example expression for limiting a value in the chart to a maximum of 200:



By adding a point label, the chart displays the real value as a label to all the data values that were capped at 200:



### Horizontal Bar Charts

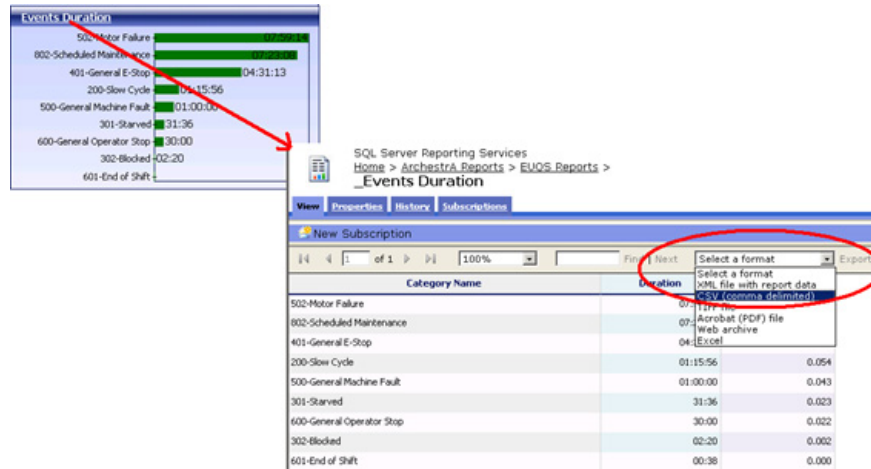
When the number of records vary greatly for a horizontal bar chart, it can be preferable to use a table object in which you embed a chart object that renders a single value. The procedure is explained at [SSW Rules to Better SQL Reporting Services 2005](#).

### Data in Charts is Not Exported

RS 2005 does not export data from charts. If a report contains a chart and you export to CSV, no data is exported. Exporting to Excel format exports the report as a series of images.

Users often want to export the data to another tool and perform further analysis. One option is to create another report file, which is simply a report data dump into a table. The data is collected from the chart.

You can make this report available as a link from the title block of the chart. Clicking the chart title in the main report calls the secondary report that lists all the data composing the chart. You can then export the data to CSV or Excel and do further analysis.



Basically, the same Dataset used for the chart is also used for the drill-down table, passing also the same set of parameter values, if applicable.

## Configuring the Report Layout

A report can often be divided in logical sections, which typically contain a single chart, table, matrix, etc. The section often has a title block and possibly other components such as an image, independent of the main object.

### Using Rectangles as Report Sections

All the components of the report section can be grouped together with the use of a rectangle object. Movement, duplication or deletion of a report section is greatly simplified since everything is implicitly selected by clicking on the rectangle object.

Procedure: create a rectangle object and drag and drop the section objects into the rectangle.

### Check the Print Layout

It is important when planning and designing a report to size the report sections with respect to page breaks.

When possible, put chart sections (fixed height) before table sections that can be of variable length. It is also important to verify the height of a chart: sometimes only a few millimeters are missing on a page and a chart is therefore moved to the

next page, leaving a lot of empty space at the bottom of the page. Decreasing a chart height by a few millimeters may avoid a page break.

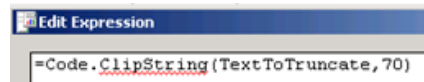
To verify the page layout of a report, you may either print it or export it to PDF format and check the resulting PDF file layout.

### Text Box Length

Some report layouts are not well suited for long string lengths. You can set Text boxes to a fixed length such that any text exceeding the length of the text box is truncated.

However, it may be preferable to indicate that the text was truncated by adding an ellipsis mark (...) at the end of the string. A function to this effect was added to the report template: **ClipString**. This function truncates a string after **n** characters and adds an ellipsis mark at the end.

To use the ClipString function, use an expression as follows:



This example truncates the string to 70 characters and adds suspension point.

For consistency, you can also use the MaxTextLength report parameter to control the length of similar text boxes in a report.

### Report Colors

The report templates use a report parameter (LayoutColorCodes) to control the colors used in the report.

The report parameter extracts the color codes from a table in the aaReports database (aaRSLayoutColorCodes). This allows for a certain consistency throughout the report. It also allows modifying easily the color scheme of a report set, just by editing the color codes in the table, without modifying the report code itself.

To use the color codes defined in the table, use the following expression:



where the index of the color (12 in this case) is defined in the **Colors** section, later in this document.

## Report Contents

There are no specific rules relating to how reports are structured in terms of reports, subreports, tables vs. matrices, etc. However, reports should be structured to avoid repeating queries against the same tables within the same report execution, particularly for tables that are likely to have tens of thousands of records or more in production applications.

### Report Structure

Subreports should be used with caution when placed within a Dataset that can return many records because the query associated to the subreport will be executed as many times as there are records in the Dataset. Subreport queries are executed at the same time the main report is rendered.

### Hyperlinks

Some reports include hyperlinks within the report. The hyperlinks link to other reports, or to other content within the Wonderware Information Server (WIS) site.

All such hyperlinks must use URLs that use the same hostname, protocol, and port specified in the client browser. They must also correctly consider any installation options. In other words, they must still work when WIS is installed to a non-default virtual directory.

For example, if WIS is installed on a computer named "Foo" and a browser client accesses WIS across a NAT router using the hostname "https://foobar.acme.com/sv:8080", the hyperlinks included in the report should direct the browser to a URL in the form

```
https://foobar.acme.com/sv/____:8080
```

rather than to

```
http://foo/wonderware/____,
```

```
https://foobar/wonderware
```

```
http://foo/wonderware/____:8080
```

etc.

Note that the report server URL is available as a global variable within RS as `Globals!ReportServerUrl`.

The WIS URL is available in the system parameters list.

Some reports use "hidden hyperlinks." These are standard hyperlinks, but they are formatted within the report using the same font style as other report content, not the special hyperlink "blue" and underline typical of hyperlinks.



## Report Parameters

### Date/Time Parameters with Time Picker

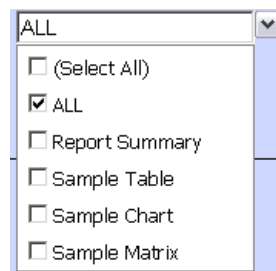
The Time picker in WIS is enabled when parameters named `StartDateTime` and `EndDateTime` are present in a report.

Report developers must be aware that the time received from the time picker is always UTC based, regardless of the selection that was made for the `TimeZoneId` parameter.

However, if the report is run with Report Manager of RS, the Time picker is not present and the times are not converted to UTC. The SQL objects handling these datetime parameters should handle both these situations, as described in the SQL Objects section below.

### Multi-valued Parameters: (Select All)

SQL Server 2005 Service Pack 2 re-introduced a special entry in the drop-down boxes of multi-valued parameters: (Select All):



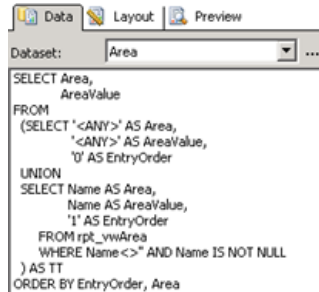
When you select the **Select All** option, the Reporting Services actually passes the complete list of values, separated with commas.

This feature works well with parameters with a few values. However, when the list is very long, the resulting parameter string becomes problematic, especially if it is passed to a stored procedure as a parameter. If the resulting string length exceeds the length of the stored procedure parameter, the string is simply truncated, which generally generates an error in the stored procedure.

As a work around to this issue, it is recommended to manually configure the ALL selection.

### To manually configure the ALL selection

- 1 In the list of available values for the parameter, add an entry such as <ALL> or <ANY>. If the list of values is defined by a query, one option is to add a UNION statement for the <ALL> or <ANY> entry.



- 2 In the query or stored procedure used to extract the data, handle the <ALL> or <ANY> label accordingly. Here is an example of a WHERE clause accounting for the <ALL> entry:

```
AND (Field IN (@Parameter) OR '<ALL>' IN
(@Parameter))
```

### Long List of Values Replacement

In some cases it may be appropriate to replace a very long list of values in drop-down boxes for parameters by a simple text box where users can enter a search pattern, which is then used as a LIKE filter in the WHERE clause of the Dataset query. This may be particularly appropriate for a list of Batch IDs for example, which is constantly growing over time.

### Query-based Default Values

One of the strengths of RS is the ability to define linked reports for which parameters can be fixed and/or overridden. In the original report, default values for parameters can be left unassigned, fixed with a constant, based on an expression, or based on a database query. Be aware, however, that one of the consequences of query-based default values is that the parameter value cannot be overridden in a linked report.

## Floating Point Default Values

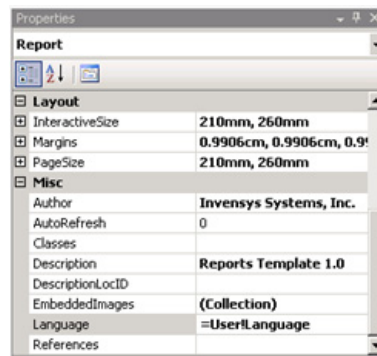
When a parameter of Float data type is assigned a value with decimals as a default, define the value as an expression by adding an equal sign in front of the value. For example, to assign a default value of 0.5 to a parameter, use =0.5 (instead of simply 0.5). Using an expression definition will ensure that the value is converted to the appropriate decimal separator for the client browser (some languages use a comma as the decimal separator).

## Preparing for Localization

### Number and date/time format

The decimal separator for floating point values, and the date and time formats are controlled by the report language, which is one of the properties of the Report object (set in the properties window in BIDS).

In order for the report to adapt to the language of the browser, use the Language property as follows:



### Currency Format

Currency format should be used with care since depending on the client browser settings, the currency symbol may differ. Therefore, numbers formatted as currency could have their currency symbol switched without any business rule applied to use an exchange rate. In general, the currency symbol should be provided by the database, possibly as a separate field from the number.

### Report Language

Labels in the report can be localized to the browser language if the report takes advantage of the aaRSLayoutLocalizedText table and the LayoutLocalizedText report parameter.

Follow the steps described in the tutorial for creating a report, earlier in this document.

Note that the scope (Scope field in the aaRSLayoutLocalizedText table) of all localized text entries for a report set should be prefixed with an acronym uniquely identifying the report set. For example, the report set associated to Equipment Performance Module 2 could be identified with the acronym EPM2. Consequently, the scope of all entries in the localized text table could be EPM2-ReportName.

## Configuring SQL Objects

### Object Naming

All SQL objects added to support the Datasets of reports should be prefixed with an acronym identifying the report set. For example, the report set associated to Equipment Performance Module 2 could be identified with the acronym EPM2. Consequently, views, functions, stored procedures, etc. associated to this report set could be called **EPM2\_SQLObjectName**.

### UTC and Time Zone Handling

Depending on whether WIS or RS Report Manager is used to execute a report, parameters named StartDateTime and EndDateTime will be UTC based or not. This situation needs to be handled by the SQL queries used for report Datasets. Here is an example:

```
DECLARE @EndDateTimeUTC DateTime
DECLARE @ActualTimeZoneId int

SET @ActualTimeZoneId = @TimeZoneId

-- Need to determine if the parameter values came from
-- the Arcestra Reports parameter page or the standard
-- Microsoft page

-- For the logic below to work correctly, the report
-- should include an integer parameter "TimeZoneId" with
-- a default value of "-1",

-- which will be overridden by the Arcestra Reports
-- parameter page

IF @ActualTimeZoneId = -1
BEGIN -- The Arcestra Reports parameter page will
      never supply a value of "-1". Assume the time is in
      "server local" time rather than UTC
      SET @ActualTimeZoneId = -100
      SET @StartDateTimeUTC = dbo.TimeZoneToUtc
        (@StartDateTime, -100)
      SET @EndDateTimeUTC =
        dbo.TimeZoneToUtc(@EndDateTime, -100)
END
```

```

ELSE
    BEGIN -- The times are already in UTC, coming from
           the ArchestrA Reports parameter page
        SET @StartDateTimeUTC = CAST( @StartDateTime AS
            datetime )
        SET @EndDateTimeUTC = CAST( @EndDateTime AS
            datetime )
    END

```

## Displaying Times in Reports

All reports must display time expressed in the time zone of the RS server. Ideally, they should also support other time zones by using the `TimeZoneID` parameter. The following sections explain some of the user-defined SQL Server functions available to handle time in report Datasets. For a more complete list and description of these functions, please refer to the WIS Administration Guide.

### Date/Time Data

All date/times used in the report are, by default, expressed in local time on the report server. SQL "datetime" database columns that are expressed in UTC are converted to "server local" time using the user-defined SQL Server function "dbo.UtcToTimeZone"

### Date/time Formatting

All date/time formatting within a report should be based on "culture-aware" formatting rather than a fixed format. This is done automatically if the report language is set to the local language as described in the section [Preparing for Localization](#) on page 35.

All instances of date/times that are used internally (e.g. as parameters, embedded in a URL, etc.) must be in a consistent, culture-neutral format (e.g. YYYY-MM-DD HH:MM:SS.SSS) and with a clear time zone (e.g. UTC or accompanied by a `TimeZoneId`).

### Data In UTC

For data that is stored in the database as UTC (even if it is also stored in a local time), use the "UtcToTimeZone" function described below. For example, if the following query would display data in UTC:

```

SELECT EventStampUTC, EventDescription
FROM FooEvents
WHERE EventStampUTC BETWEEN %StartDateTimeUTC%
AND %EndDateTimeUTC%

```

Then, this should be replaced with:

```
SELECT EventStampUTC,  
       dbo.UtcToTimeZone (EventStampUTC,%TimeZoneID%)  
AS EventStamp, EventDescription  
FROM FooEvents  
  
WHERE EventStampUTC BETWEEN %StartDateTimeUTC%  
AND %EndDateTimeUTC%
```

The report should use the calculated "EventStamp" column. This approach will enable you to select any time zone for display purposes. Be sure to place the criteria on the UTC columns to avoid the need for a full table scan when executing the query.

### Data In Server Time

For data that is only stored in the database as local time on the server, use the local time column directly. For example:

```
SELECT EventStamp, EventDescription  
FROM FooEvents  
  
WHERE EventStamp BETWEEN %StartDateTime%  
AND %EndDateTime%
```

Do not use the "TimeZoneID" parameter because if the browser is in a different time zone than the server, the data returned will not be in the expected time zone (which should default to the browser's time zone, not the server's).

While you can use the time zone conversion functions, be careful to avoid queries that cause a full table scan. For example, if you want to include a UTC column and UTC criteria, do the following:

```
SELECT EventStamp,  
       dbo.TimeZoneToUtc (EventStamp,-100) AS EventStampUTC,  
       EventDescription  
FROM FooEvents  
  
WHERE EventStamp  
  
BETWEEN dbo.UtcToTimeZone (%StartDateTimeUTC%,100)  
AND dbo.UtcToTimeZone (%EndDateTimeUTC%,-100)
```

Specifically, do not do the following:

```
SELECT EventStamp,  
       dbo.TimeZoneToUtc (EventStamp,-100)  
AS EventStampUTC, EventDescription  
FROM FooEvents  
  
WHERE dbo.TimeZoneToUtc (EventStamp,-100)  
BETWEEN %StartDateTimeUTC% AND %EndDateTimeUTC%
```

## Decode Values

When appropriate, columns that hold a code or index (e.g. the integer "TagType" column in the InSQL "Tag" table) should be converted to a string (e.g. "Analog", "Discrete", etc.).

## Schema Usage

All database objects should be created under the "dbo" schema and should not explicitly use a schema in the references to database objects.

For example, use:

```
SELECT * FROM Foo"
```

rather than

```
"SELECT * FROM dbo.Foo".
```

## Naming Conventions

Database views that directly match tables and views in the linked server should exactly match the name of the database object in the linked database.

Views that are specific to a report set should have a meaningful name (as specified in points 1 to 3 below). In all cases, add a 3 or 4 letter prefix identifying the report set for which the object is created.

Columns (and for stored procedures and functions, parameters), that directly mirror columns (or, parameters) in the linked database should exactly match those in the linked database.

Columns (and for stored procedures and functions, parameters), that are specific to objects in the reporting database should follow the conventions below.

- 1 Words used within names should be complete words rather than abbreviations.
- 2 Use mixed case with initial capital letters for each word (e.g. AValidColumnOrParameterName) and do not include underscores.
- 3 Names that reflect plural items or relationships should use plural names.
- 4 Identifier fields should include the suffix ID (in this exact case; not Id nor id). Example: TextID.
- 5 Unique Identifier field names should include the suffix GUID (in this exact case).
- 6 Names should be organized as "NounAdjective", including the noun first to simplify browsing a sorted list of columns/parameters. For example, "AlarmTimeOriginal" rather than "OriginalAlarmTime".

- 7 Columns/parameters that are date-time values should include "DateTime" in the name if it represents local time on the client or on the server and, instead, include "DateTimeUTC" in the name if it represents UTC time.

## Report Templates Description


Reports should be derived from the 2 templates available in WIS: one portrait template and one landscape template. These templates include the common infrastructure between all reports including:

- Parameter definitions
- Dataset definitions
- Header and Footer objects
- Sample report sections

Report objects within the templates are defined with the standard title bars and color schemes described later in this document. It is therefore recommended to copy and modify these objects as needed when creating report sections, as explained in the procedures earlier in this document.

The following section describes the layout considerations of the templates and reports.





## Sample Report Template

Archestra Reports

---

**Filter Criteria**

**Time Period:** 1/10/2008 12:00:00 AM - 1/11/2008 12:00:00 AM [ (GMT-05:00) Eastern Time (US & Canada) ]

**Tag Name:** **Group:**

**Report Sections:** ALL

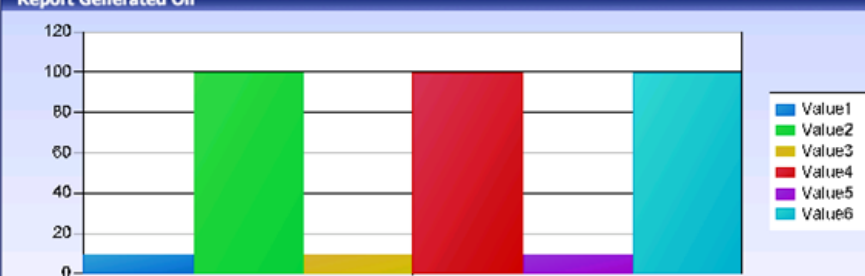
---

**Sample Table**

Scope	Text Id	Language Code	Localized Text	Description
ReportTemplate	0	en-us	Report Title	Title of report in header
ReportTemplate	1	en-us	Report Generate	Label for report execution date in footer
ReportTemplate	2	en-us	Page	Label for page numbers in footer
ReportTemplate	3	en-us	Filter Criteria	Title bar text for Report Definition Summar...
ReportTemplate	4	en-us	Report Sections	
ReportTemplate	5	en-us	Time Period	
ReportTemplate	6	en-us	Duration	
ReportTemplate	7	en-us	Time Zone	
ReportTemplate	8	en-us	Tag Name	
ReportTemplate	9	en-us	Group	

---

**Report Generated On**



---


**Sample Matrix**

	0	10	20	30	40
0	0	10	20	30	40
1	1	11	21	31	41
2	2	12	22	32	42
3	3	13	23	33	43
4	4	14	24	34	44

---

Report Generated On 1/11/2008 11:13:56 AM

Page 1 / 1



## Colors

The following table lists the standard colors used in the reports. These color codes are defined in the **aarsColorCodes** table and are available as a report parameter set within the report templates. The parameter is named **LayoutColorCodes** and is referenced by using the following syntax:

```
Parameters!LayoutColorCodes.Value(index)
```

where index is the color index as defined in the table below. All objects in reports should be colored based on this parameter set. This will ensure consistency between the reports and avoid modifications to reports source code when color schemes are modified.

Color Index	Color Use Description	R	G	B	RS Color Code
0	Table odd row background	251	251	251	#FBFBFB
1	Table even row background	239	238	234	#EFEEEE
2	First alternate even row background	229	240	243	#E5F0F3
3	Second alternate even row background	237	235	245	#EDEBF5
4	Third alternate even row background	230	242	227	#E6F2E3
5	Fourth alternate even row background	246	234	226	#F6EAE2
6	Primary foreground	17	30	98	#111E62
7	Alternate foreground	0	0	0	#000000
8	Internal column right border	214	214	214	#D6D6D6
9	Internal column left border	255	255	255	#FFFFFF
10	Chart background gradient start	205	214	249	#CDD6F9
11	Chart background gradient end	238	241	255	#EEF1FF
12	Primary background	219	226	251	#DBE2FB
13	Section header background	44	58	134	#2C3A86
14	Section header foreground	255	255	255	#FFFFFF
15	Footer dividing line	154	154	202	#9A9ACA

## Page Size

Unless there is a definite requirement for larger paper size, all reports should be on a page size of 210mm x 279mm (the minimum of A4 & U.S. Letter with a margin) either portrait or landscape. They must have a margin of at least 10mm on all sides, otherwise some printers will truncate the contents of the report..

## Page Header

The Report header contains the Report name and is placed in the upper right corner of the report (Tahoma 14 pt bold).

Site location name is shown below the Report Name separated by a line of 1pt (Tahoma 10 pt bold). This name is common to all reports and dynamically loaded from the database on the reports server via a report parameter.

The Report header also includes a logo at the top left corner that is dynamically loaded from a URL on the reports server to facilitate modifying the image without modifying the report definition itself.

The Report Header has no border and no background color. Text and line are using the "Primary foreground" color. The height of the header section is 15 mm.

## Page Footer

The Report footer includes the date the report was printed, the page number for each page, and the total number of pages in the report.

At the bottom right of the Report footer the Invensys Wonderware logo appears.

The height of the footer section is 9 mm.

## Reports Sections

A report contains at least two sections: a "Report Definition Summary" section and one or more data region. Data regions are rendered as a table, list, matrix or chart. Each report section should be contained within a rectangle object. Objects within the report section are included in the rectangle, which makes it easier to keep the objects within the report section together. It also facilitates report rendering (page break control) of each report section.

It is generally preferable to avoid the placement of two report sections side by side unless the height of these report sections is fixed (charts or other data regions with fixed record count).

Otherwise, when the height of the report sections is variable, tests should be made to verify how the layout is actually rendered.

Report sections have a solid border of the primary foreground color with 1 pt width. The default background of report sections is set to the primary background color.

The body of the report is also set with the same border and background as a report section.

## Report Section Headings

Each section of the report have a section header with text in Tahoma 10 pt bold. The background of the heading is an opaque image that repeats for the length of the section heading. Note: these headers still use a background color for legibility within Report Designer.

## Report Definition Summary Section

The values of the selected report parameters (indicated in each report) are included on the first page of the report in a "Report Definition Summary" section. When applicable the summary also includes the "display name" of the time zone by using the user-defined SQL Server function.

```
dbo.TimeZoneDisplayName (-100)
```

or

```
dbo.TimeZoneDisplayName (%TimeZoneId%)
```

Data Filtering Criteria			
Start DateTime:	2006-05-21 16:00:00	End DateTime:	2006-05-21 16:00:00
Area:	Area1, Area2		
Product:	Product1		

## Tabular Sections

Tabular sections of reports use Tahoma 9 Product Tracking. Header rows are in bold, using the primary foreground color, and with a background color of the even rows for the column. The actual data rows use normal font weight and the alternate foreground color. For tables/matrices that include a header column, it is set to bold. Except as indicated, all columns use the standard odd/even row background colors. The borders between columns of the same section use the indicated colors for the right and left borders. The outer borders of the table use the standard foreground color.

## Matrix Sections

Matrix sections follow the same rules as for tabular sections in terms of font size and background colors. Care must be taken when defining the column size of dynamic columns in a matrix; if the number of columns exceeds the page width, the matrix will continue on another page.

## List Sections

List sections are basically free format sections where report items are dropped. The background of the list is set to the primary background color.

## Chart Sections

Chart sections are set with a top-bottom gradient using the "Chart background gradient start" and "Chart background gradient end" colors.

## Units of Distance

Within reports, all distance measurements are in "millimeters" and use a consistent grid size defined in "millimeters". This may involve changing the regional settings of the development station to use metric units instead of the U.S. measurement system.

## Testing Your Reports

This section provides a few items to verify when testing a report. This is not an exhaustive list but rather a starting point to ensure a certain consistency between reports.

### Verify Data Returned by SQL Objects

Verify all SQL objects created for a report set by comparing the contents of the original data source with the SQL objects. Some items to consider are:

- Is the number of records consistent with the source?
- Were there some records eliminated by an INNER JOIN?
- Were there additional records created by an OUTER JOIN? This is especially important when grouping data and computing summary values.
- Are NULL values handled correctly?
- What is the execution plan for the report query? Are indexes missing?

### Check Calculations

Check all calculations, whether they are in the query or in report expressions. It is always a good practice to verify these calculations by hand using a representative Dataset and compare with the report output. Some items to consider are:

- Are NULL values handled correctly in calculations?
- Be careful with Datasets returning non-additive summarized values such as averages or ratios. If these values are in turn summarized in a chart or a table grouping, the results may not be correct. It is sometimes necessary to return additive summarized values such as sum and count, to compute an average in a chart or table grouping.

### Check Report Parameters

#### Query Based Parameter Contents

Parameters making use of queries to populate the list of available and/or values should be verified for contents. Make sure the query returns all expected data.

### Parameter Dependency

If some parameters depend on the selection of another parameter, make sure that the behavior and content of the dependent parameter is as expected.

### Data filtering parameters

Parameters that filter a Dataset used in the report should be verified. Some items to consider are:

- If a single value is selected in the parameter, is the resulting Dataset correct?
- If multiple values are selected (if allowed), is the resulting Dataset correct?
- If <ALL> is selected, is the resulting Dataset complete, including records with null values?
- Free text parameters allowing to enter a filter mask are working as expected?
- For each of the above tests, check that the Data Filtering section at the top of the report (if applicable), displays the parameter selection adequately.

### Parameter Behavior

Check all other parameters in the report to make sure that they work as expected. Check all possible options.

## Check Drill-down Paths

Check all report actions that drill-down to another report or web page. Some items to consider are:

- Check that the target is correct. If the target is a URL, make sure that the root of the URL is as specified in Hyperlinks on page 32.
- Check that the parameters passed to the target are correct.
- When applicable, check that the reference of the clicked object is passed correctly to the target. For example, if you click on a bar chart to drill-down to a detailed report, is the target report filtered based on the clicked bar?

## Check Report Localization

If the report is localizable, you should check the report using different browser languages. You can change your browser language in Internet Explorer by selecting Tools, Internet Options, Languages. Make sure to set the language you want to use at the top of the list if several languages are available.

For .NET controls (such as the Arcestra Reports parameter page Tag Picker/Time Picker) the language comes from the Regional Settings at the time the browser was launched. Changing the settings will require restarting the browser for them to take effect.

You should choose languages with different datetime, currency and number formats. Some items to consider are:

- Date and time formats in the report: are they as per selected browser language?
- Date parameters: verify that there is no confusion between mm/dd and dd/mm using dates with days smaller or equal to 12. Verify that the dates were interpreted correctly in the report output.
- Check that there is no unexpected currency symbol change.
- Check that labels are translated correctly as per input in table aaRSLayoutLocalizedText.

## Check Printed Report Format

Report layout should be verified both on the screen and printed on paper (or PDF format). Some items to consider are:

- Are there some objects too wide for the paper size, causing them to overflow on two pages horizontally?
- Are the report header and footer displayed correctly, both in terms of content, layout and position?
- Are the page breaks occurring as expected?
- If a table object spans a second page, are table headers repeated correctly?

## Check Expand/Collapse Behavior

If the report includes objects that expand and collapse using +/- (or other symbols), you should check the consistency of the symbol with the actual expand/collapse behavior. Also verify the initial display of the expansion symbol. If a report parameter drives the initial state of the expansion, verify that the initial symbol corresponds to the expansion state.



## Check Color Application

Check that the use of colors is consistent throughout the report. In particular if alternating colors are used in tables, verify the colors of rows and columns. It is also important to verify the actual printout of the report on a grayscale printer.

## Check Chart Limits

If limits were applied to charts, verify with appropriate data that the chart does actually limit values and display a marker label for those values, if applicable.

## Check Time Zone Selection

Check that times displayed in the report are consistent with the parameter values. Some items to consider are:

- Make sure that there is no confusion between UTC and local time. This may involve setting datetime parameters to specific value and check that the resulting Dataset only includes the expected records.
- Using WIS and the time picker, change the time zone selection and check that the report actually reflects that selection
- Using RS Report Manager, make sure that the time used is that of the server.



# Index

## A

- associating dataset fields to table columns 20
- associating the report section to your dataset 19
- associating the table to your dataset 19

## C

- chart rendering 28
- chart sections 45
- checklist
  - calculations 46
  - chart limits 49
  - color application 49
  - drill-down paths 47
  - expand/collapse behavior 48
  - printed report format 48
  - report localization 47
  - report parameters 46
  - time zone selection 49
- colors 31, 42
- configuring SQL Objects 36
- configuring the report layout 30
- creating a dataset for the view 18
- creating a new project 12
- creating a new report 10
- creating ArchestraA reports 9

- creating new Visual Studio and Report Server projects 12
- creating SQL Objects and Report Datasets 17

## D

- displaying times in reports 37
- displaying user selections for report filtering 25
- document overview 10
- documentation conventions 7

## G

- glossary of terms 9

## L

- list sections 45

## M

- manually configuring the ALL selection 34
- matrix sections 45
- modifying the report template 16

## O

- overview 10

## P

- parameters

- date/time with Time Picker 33
- floating point default values 35
- multi-valued 33
- query-based default values 34
- values replacement 34
- preparing for localization 35

## R

- references 10
- reports
  - contents 32
  - definition summary section 44
  - hyperlinks 32
  - page footer 43
  - page header 43
  - page size 43
  - parameters 33
  - section headings 44
  - sections 43

- templates description 40

## S

- switching to metric units 11

## T

- tabular sections 44
- technical support, contacting 7
- testing and preparing for deployment 26
- testing your reports 46
- tips and guidelines 28

## U

- units of distance 45
- using metric units 11

## V

- verifying data returned by SQL Objects 46