Microsoft Project Server 2010 Reporting with Excel Services

Written by: Jean-Francois LeSaux, Microsoft Services

Steven Haden, Microsoft Services

Contributor: Amit Abrol, Microsoft Services

Microsoft

Information in the document, including URL and other Internet Web site references is subject to change without notice. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you a license to any patent, trademarks, copyrights, or other intellectual property that are the subject matter of this document.

Table of Contents

© Microsoft Corp. All rights reserved

1. Foreword

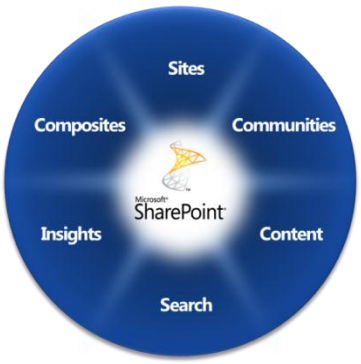
This document is part of a series of papers covering Microsoft Project Server 2010 Reporting. The following papers have been already published or will be published along with this document.

'Reporting with Microsoft Project Server 2010': http://technet.microsoft.com/en-us/library/gg188101.aspx

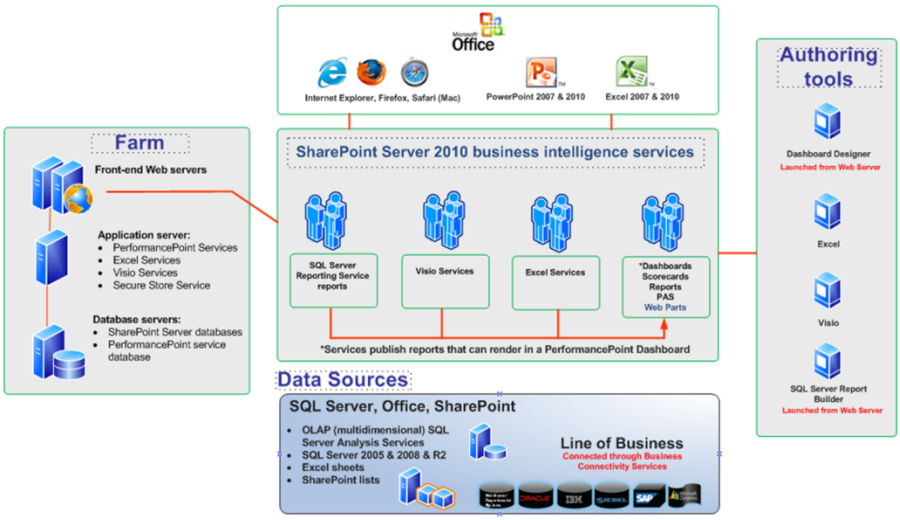
'Creating Dashboards for Microsoft Project Server 2010'

2. Introduction

Microsoft Project Server 2010 has brought many enhancements to a new architecture that was released with Office Project Server 2007. The one enhancement with the biggest impact on the architecture has been the requirement for Microsoft SharePoint Server 2010 Enterprise Edition. This change of platform requirement has brought a myriad of features which are now available to Project Server 2010 such as BI Insights (Business Intelligence), Communities, Sites, Composites, Content and Search …



One of the biggest benefits for Project Server 2010 is, without contest, the availability of all the BI Insights services composed of Excel Services, PerformancePoint Services, Visio Services, SQL Server Reporting Services (in SharePoint integrated mode) and Office Web Apps.



For an overview of the BI Insights offerings, refer to this downloadable poster: http://www.microsoft.com/downloads/en/details.aspx?FamilyID=fc97d587-ffa4-4b43-b77d-958f3f8a87b9&displaylang=en

Technical Architecture

Our Business Intelligence features leverage Excel Services in Microsoft SharePoint Server 2010 as the base functionality because most people use Microsoft Excel to visualize data and it is a tool that many people already know how to use.

There are four core components to this solution.

Excel client. The Excel client is used to author and publish new reports. This solution works with Excel 2007 SP2 or later.

Office Data Connections. Office Data Connections (ODC) are used to store the connection information, the SQL Query and the Secure Store Target Application ID. External ODCs are used to allow you to manage data connection and query information externally to the reports that consume the data. These two components together are the deliverables from the report author.

When you provision a new Project Web Application site or when you create a new OLAP database, ODCs and attached templates are automatically generated in the Business Intelligence Center.

Excel Services. Excel Services provides rendering and interactivity support on the web. This service enables the user to share reports easily with others. It also enables a user to filter the data in a report dynamically to meet a particular need.

Secure Store. Secure Store is a SharePoint Server service that is used to store credentials in a Target Application Profile. These profiles help avoid double-hop authentication situations and provide control around who has access to what data for a given Target Application Profile. In SharePoint Server 2007, this service was known as Single Sign-On service, or SSO.

This diagram explains the architecture behind Excel Services and the different components present in SharePoint Server 2010.

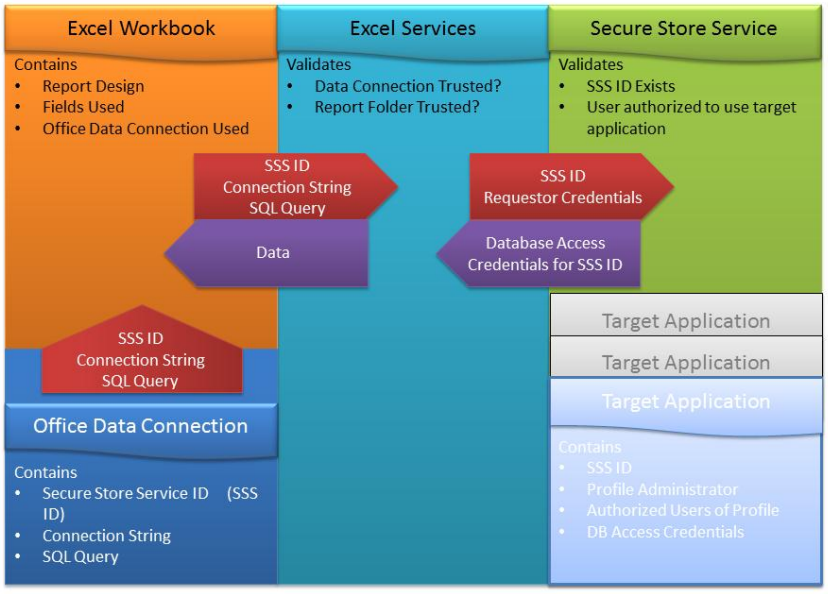


Figure 1 - Reporting Architecture of Project Server 2010

The diagram illustrates the interactions between the four components. The arrows denote what information is passed between the components and in what direction.

PerformancePoint is called out above as it is used to create the Business Intelligence Center as it is their service that provides this infrastructure. It is not used for the core reporting features. However, you can easily develop PerformancePoint reports over Project Server data.

3. Overview

The purpose of this document is to provide enough detail to be able to create and deploy reports that will support the deployment of a Microsoft Enterprise Project Management (EPM) Solution. Microsoft EPM Solution relies on Microsoft SharePoint Server 2010, Microsoft SQL Server 2008 or 2008 R2, Microsoft Project Server 2010 and Microsoft Project Professional 2010. Although each individual component of the solution is well documented on TechNet and MSDN (for Business Intelligence), little documentation exists, as of today, that describes a step-by-step approach to building reports specific to Project Server 2010 data with Excel and Excel Services.

This document is not intended to provide an in-depth description of all the features available in Excel 2007 or 2010 and Excel Services in Microsoft SharePoint Server 2010 when building a report, but rather to provide the building blocks required to 'be up and running' quickly.

A 'References' section at the end of this paper provides useful links to TechNet and MSDN resources, for additional information.

Tools

SharePoint Server 2010 and Project Server 2010 Configuration

The different components that are involved in Reporting with Project Server 2010 are illustrated in this figure:

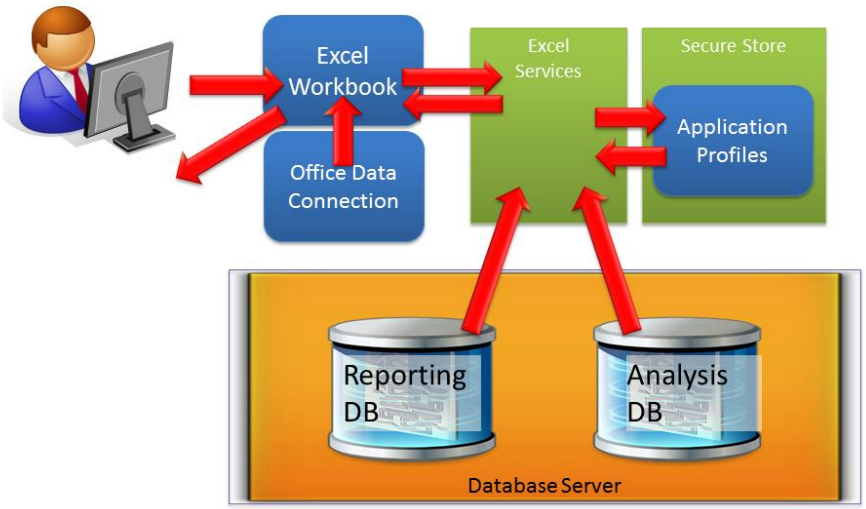


Figure 2 - Configuring Reporting in Project Server 2010

On the client side we have:

- Microsoft Excel 2010 or Excel 2007 for authoring the reports

On the server side we have:

- Excel Services
- Secure Store with an application profile that stores account credentials
- Excel workbooks and ODC
- The SQL Server Reporting database
- The OLAP database that contains the cubes

Initial Setup

After the initial configuration of Project Server 2010, the following steps have to be done for the Reporting configuration:

- OLAP database and Analysis Services configuration:
  http://technet.microsoft.com/en-us/library/ee662108.aspx

  - Add the Farm Administrator account to the OLAP users local group.
  - Configure the Farm Administrators account to have administrative permissions in SQL Server Analysis Services.

- Configure reporting for Project Server 2010:
  http://technet.microsoft.com/en-us/library/ee662106.aspx

  - Add a logon for the report authors group in SQL Server
  - Install SQL Server 2008 Analysis Management Objects (AMO)
  - Start Excel Services
  - Create an Excel Services service application
  - Configure Excel Services settings
    - Configure a trusted file location for the Templates library
    - Configure a trusted file location for the Sample Reports library
    - Configure trusted data connection libraries (one per language)
  - Start the Secure Store Service
  - Configure Secure Store Service settings
    - Create a Secure Store target application
  - Populate the Report Authors and Report Viewer Active Directory Groups
  - Configure Business Intelligence Center access
    - Grant permission to external report viewers

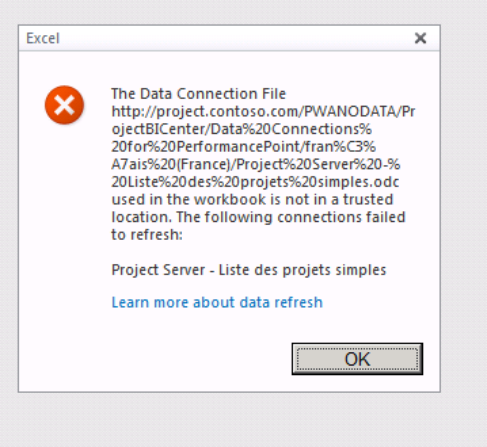
See the TechNet reference document for the detailed steps.

For each new Project Web App site created

Each time a new Project Web App site is created, the following steps will need to be performed:

- Configure a trusted file location for the Templates library
- Configure a trusted file location for the Sample Reports library
- Configure trusted data connection libraries (one per language)

If you forget to do it, you will get this error message:



Excel and Excel Services

Using Excel Services for reports allows for a variety of sharing scenarios.

1. Data can be shared via web interface using Excel Services
2. Data can be distributed via XLSX files via email or some other facility and using Excel client for viewing
    a. Files can be static snapshots of the web view, allowing you to share broadly or to modify the data for your own uses
    b. Files can also remain data-connected, allowing the user to refresh the data as needed, using the client. Note that this functionality requires Report Author level security, with a direct access to the databases.

The organization can build its own reports by using the Office Excel 2007 or Excel 2010 client and publish them to PWA BI Center.

Data Sources

You can choose two kinds of Project Server related data sources for your reports:

- Relational (Project Server Reporting DB)
- Decisional (14 Project Server Analysis Services OLAP Cubes)

Here are some criteria to help you choose from one source or another:

Relational:

- Up-to-date data
- Superset of data available in the AS cubes
- No drilldown required in report

Decisional:

- Aggregation across dimension(s)
- Efficient for reports on time phased data
- Key Performance Indicators

It is not recommended to mix these two type of data sources in the same report, because the Reporting data is always up-to-date and the Decisional data is only up-to-date when the OLAP database has been processed, which may occur only daily or weekly depending on your configuring choices.

The data itself can be separated into two main categories: OLAP and non-OLAP data. Generally, SQL Server tables are suitable for providing a snapshot of the project or resource data as it is right now. OLAP data is more appropriate for providing a snapshot of the data right now, and then allowing a comparison of the same data over time – for instance, resource availability over the next three months, or the actual cost of all projects on a monthly basis for the last six months.

The OLAP data is kept in an instance of SQL Server Analysis Services and may be consumed via any number of reporting tools. The non-OLAP data is stored in the SQL Server instance.

Relational Tables

Project Server data is stored in a set of four SQL Server databases: Archive, Draft, Published, and Reporting. Microsoft only documents and supports developing reports using data from the Reporting database. For a detailed description of the Project Server 2010 Reporting database schema, refer to the Project Server 2010 Software Developer Resource Kit available here: http://www.microsoft.com/downloads/en/details.aspx?FamilyID=46007f25-b44e-4aa6-80ff-9c0e75835ad9&displaylang=en

Tips and Tricks: Understanding the Reporting Database Schema

The Project Server 2010 SDK documentation comes with a Reporting Database Schema Help file and a neat utility consisting of a PivotTable to help select fields in the Reporting database (**RDB_Field_Selector.xlsx**). So there is no need to read all the documentation to locate a specific field in the Reporting database. See below.

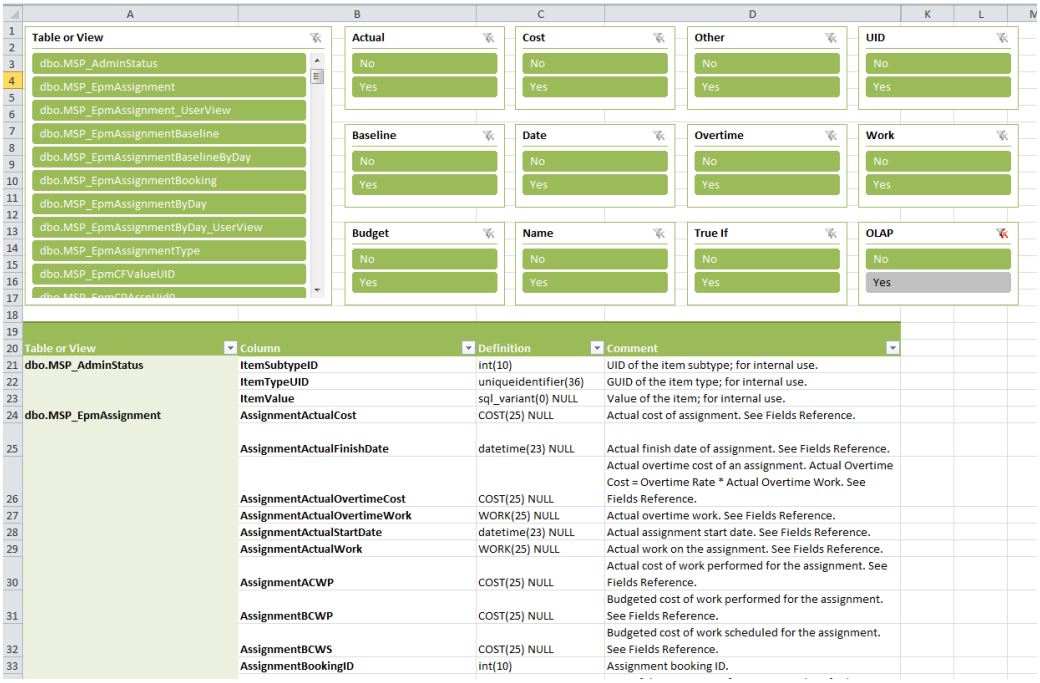


Table or View	Column	Definition	Comment
dbo.MSP_AdminStatus	ItemSubtypeID	int(10)	UID of the item subtype; for internal use.
	ItemTypeUID	uniqueidentifier(36)	GUID of the item type; for internal use.
	ItemValue	sql_variant(0) NULL	Value of the item; for internal use.
dbo.MSP_EpmAssignment	AssignmentActualCost	COST(25) NULL	Actual cost of assignment. See Fields Reference.
	AssignmentActualFinishDate	datetime(23) NULL	Actual finish date of assignment. See Fields Reference.
	AssignmentActualOvertimeCost	COST(25) NULL	Actual overtime cost of an assignment. Actual Overtime Cost = Overtime Rate * Actual Overtime Work. See Fields Reference.
	AssignmentActualOvertimeWork	WORK(25) NULL	Actual overtime work. See Fields Reference.
	AssignmentActualStartDate	datetime(23) NULL	Actual assignment start date. See Fields Reference.
	AssignmentActualWork	WORK(25) NULL	Actual work on the assignment. See Fields Reference.
	AssignmentACWP	COST(25) NULL	Actual cost of work performed for the assignment. See Fields Reference.
	AssignmentBCWP	COST(25) NULL	Budgeted cost of work performed for the assignment. See Fields Reference.
	AssignmentBCWS	COST(25) NULL	Budgeted cost of work scheduled for the assignment. See Fields Reference.
	AssignmentBookingID	int(10)	Assignment booking ID.

Project Server 2010 RDB Field Selector

OLAP Cubes

OLAP cubes provide the capability to analyze over time period, roll up and drill down into Project Server data. The 14 OLAP cubes provided natively contain data extracted from the Project Server Reporting database. Within Project Server 2010, each department may have its own set of OLAP cubes — which may then be surfaced by using Excel, Visio, PerformancePoint, and other reporting tools. OLAP cubes

are configured in PWA Server Settings, and once configured, may be accessed by using any of the reporting tools identified above.

Note that any custom fields developed by the organization must be added to the OLAP cube configuration through the standard user interface and the cube must then be rebuilt before those fields are available for report writing.

Tips and Tricks: OLAP Cubes

Excel allows users to add custom calculated fields to most PivotTables. When the PivotTable is created from a SQL Server Analysis Services connection, however, that feature has been disabled. This affects the user ability to add data calculations to reports. For example, a user exports the time-phased actual work and work fields to an Excel PivotTable, then attempts to calculate remaining work over time, in essence creating a burn down chart. That calculated remaining work column may not be created as part of the PivotTable as the source data is tied to a SQL Server OLAP cube.

To enable calculated fields in an Excel PivotTable, users may download and install the free OLAP PivotTable Extender tool from CodePlex: http://olappivottableextend.codeplex.com/

Data Connections

Project Server 2010 is shipped with default Office Data Connections (ODC) files that can be used once the configuration is complete. It is also possible to create new Data Connections from within Excel when you are authoring reports.

There are also preconfigured blank templates provided that are connected to the included ODCs to help users create new reports quickly. For each OLAP database, a blank template and accompanying ODC is created for each cube within the OLAP database.

Each time you create a Project Web App site, the following Office Data Connection (ODC) files are available in the Business Intelligence Center:

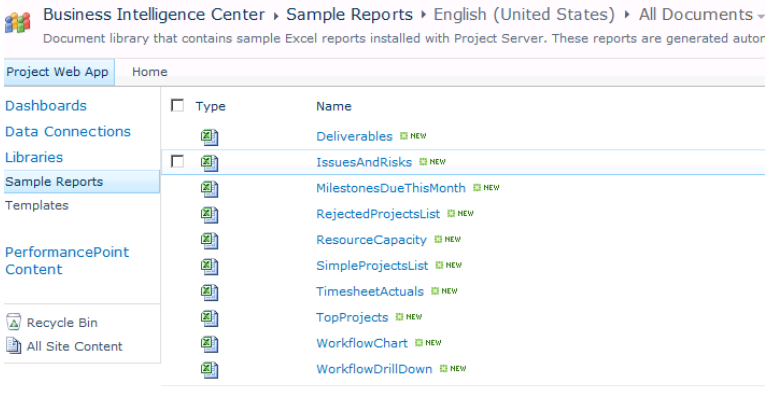
Data Connection Files:

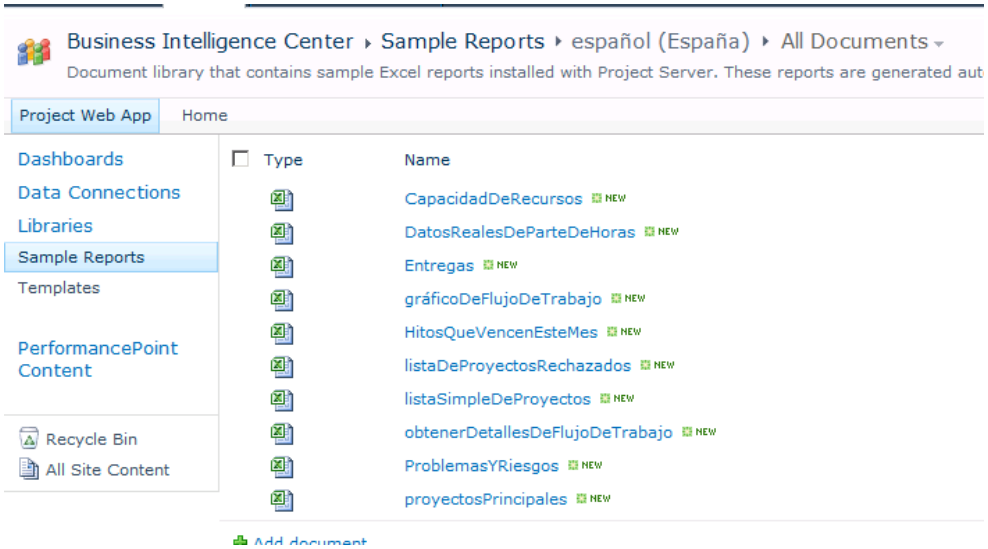
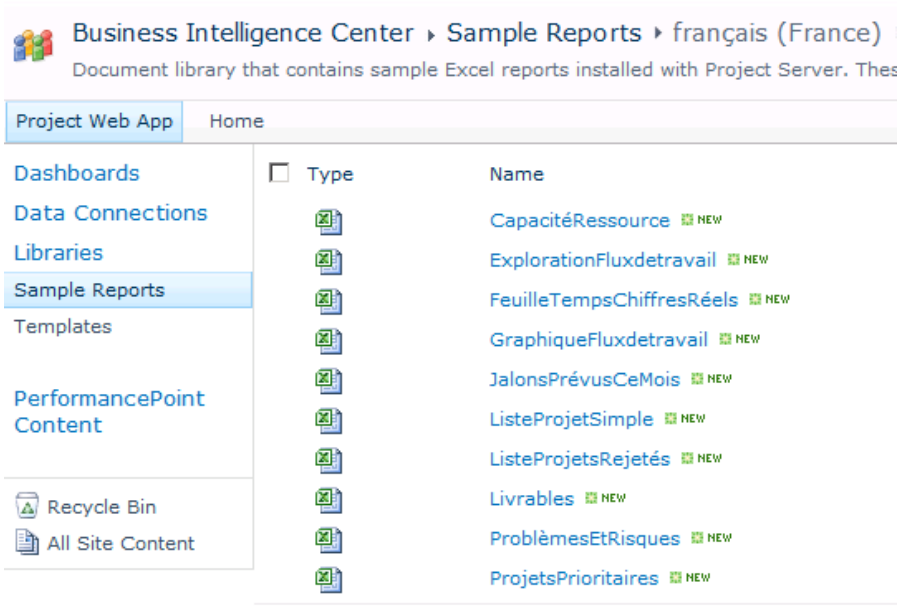
- Enterprise Project Management
  - Simple Project List (SimpleProjectList)
  - Milestone due This Month (MilestonesDueThisMonth)
  - Resource Capacity (ResourceCapacity)
- Timesheet

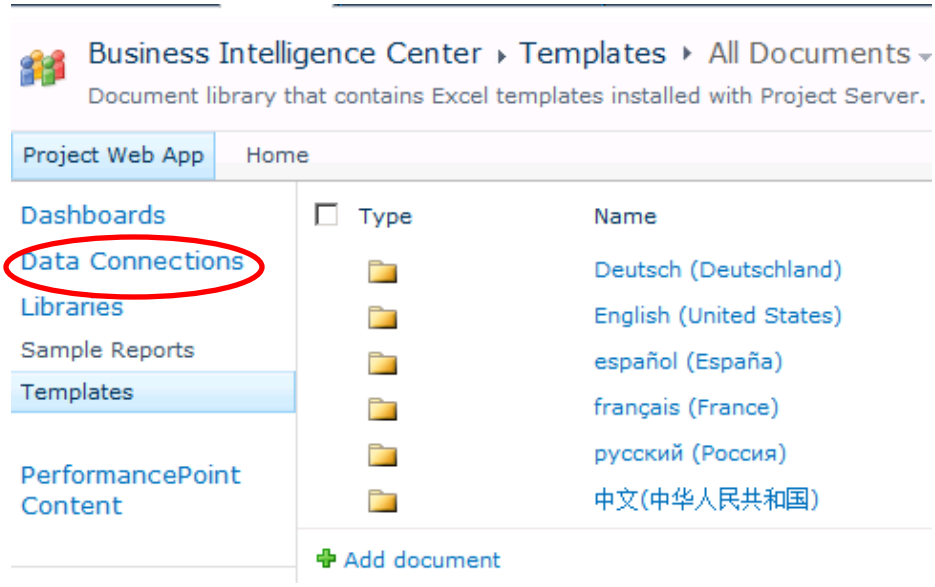
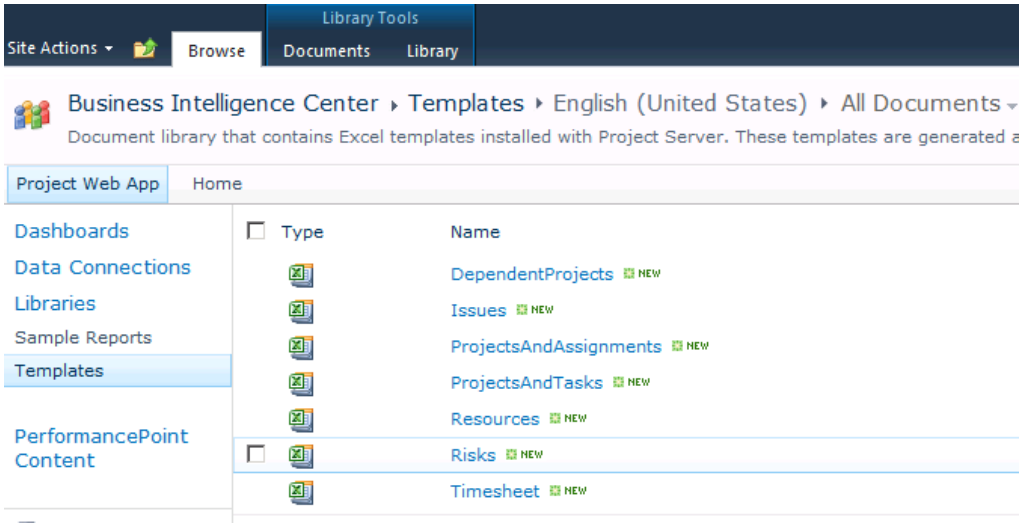
    - Timesheet Actuals (TimesheetActuals)
- SharePoint Lists
    - Deliverables
    - Issues And Risks (IssuesAndRisks)
- Portfolio
    - Rejected Project List (RejectedProjectList)
    - Top Projects (TopProjects)
    - Workflow Chart (WorkflowChart)
    - Workflow Drill Down (WorkflowDrillDown)

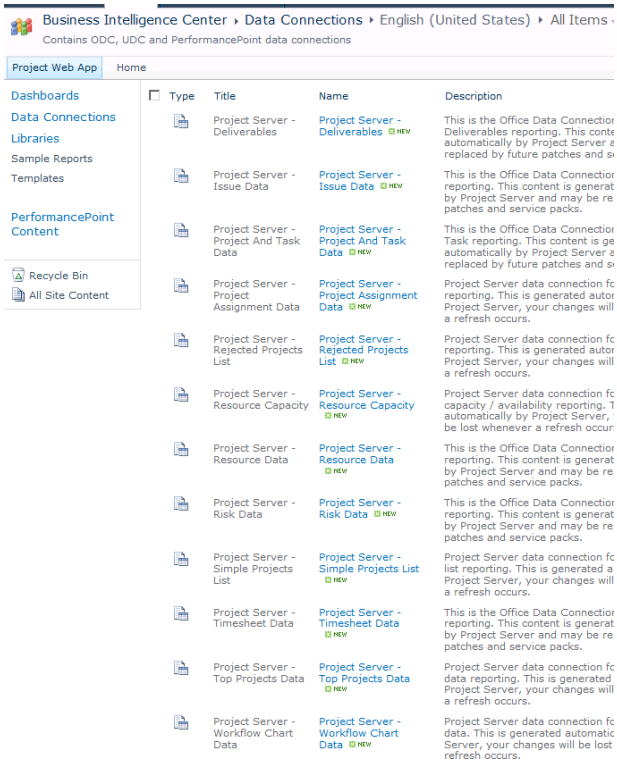
The following screen shots present how you can access these reports that come with the product:

Actions	Screen
Navigate to Project Web App Business Intelligence Center This is a SharePoint site that is a sub site of the PWA site.	

Under **Libraries** on the Quick Launch are links to two default document libraries that are created during provisioning of PWA. They are **Sample Reports** and **Templates** libraries. Here are the Sample Reports with some different language packs installed: • Deutsch • English • Español • Français • Russian • Japanese	
To access a report (once Excel Services is configured), drill into the language folder. For English (United States) You see the list of reports	

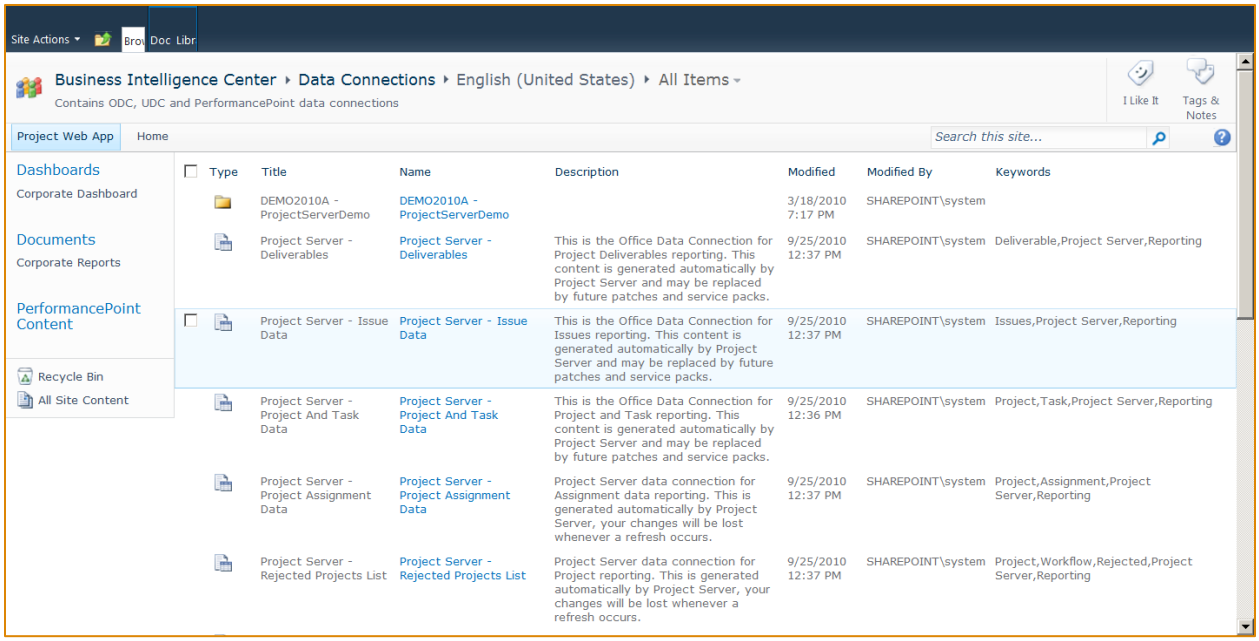
For Spanish	
And for French	

If you navigate to the Template folders Templates with the different language packs installed.	
For English (United States)	

Associated with each of these templates or sample reports there are predefined Office Data Connections (ODC) files	

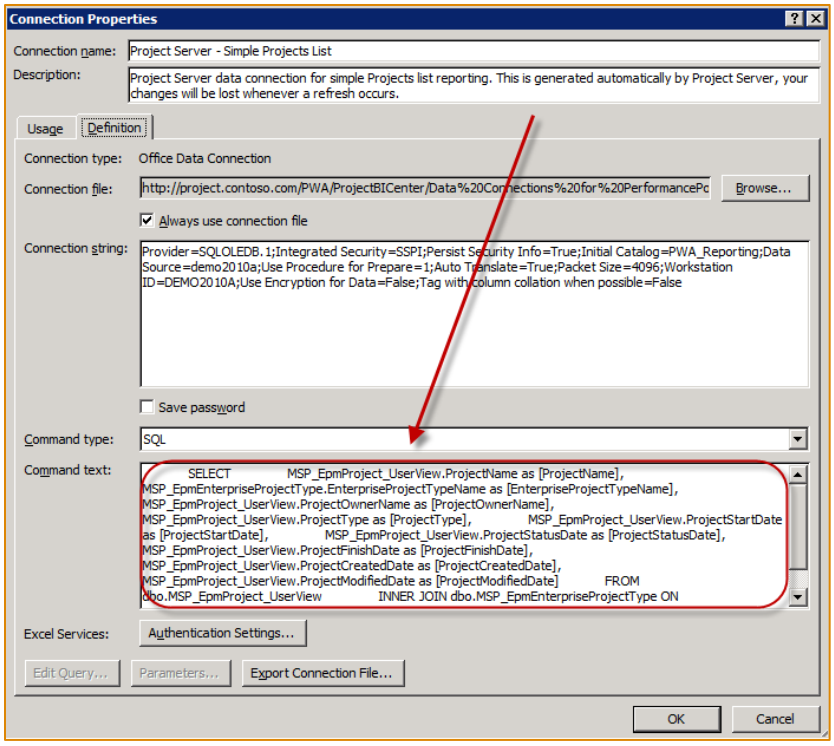
ODC files simplify the report authoring process by allowing personnel with technical skills to collect specific columns from any number of database tables, add filters and joins as required, and then provide the data sets for less technical personnel to develop reports as needed.

Project Server comes with a number of sample ODC files located in the Business Intelligence Center. Each of these ODC files may be copied and then customized to the needs of the organization. Report authors may also open the ODC files within Excel and review the settings to identify appropriate syntax and query structure.



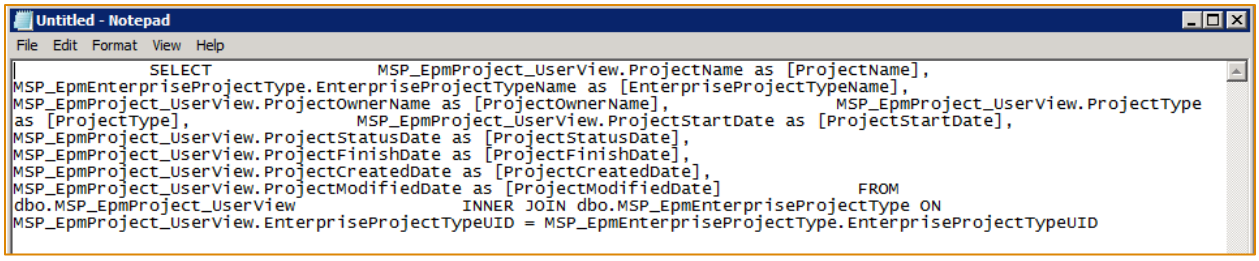
Default ODC Files

To edit an ODC file, select the ODC file to use as a basis for the report. The selected file will open in Excel. Once the data is displayed, select the Connections option from the Data tab. The ODC file is displayed. Select the Properties button to review the specific settings. This displays the Connection Properties dialog box. In the next dialog box, select the Definitions tab roughly in the middle of the box. The screen now displays the data selected to be included in the ODC file.



Identifying the Default ODC Settings

To edit the selected fields, add any new custom fields, or to implement a filter, copy the Command Text box and paste it into Notepad for editing. Update the query by using SQL query syntax.



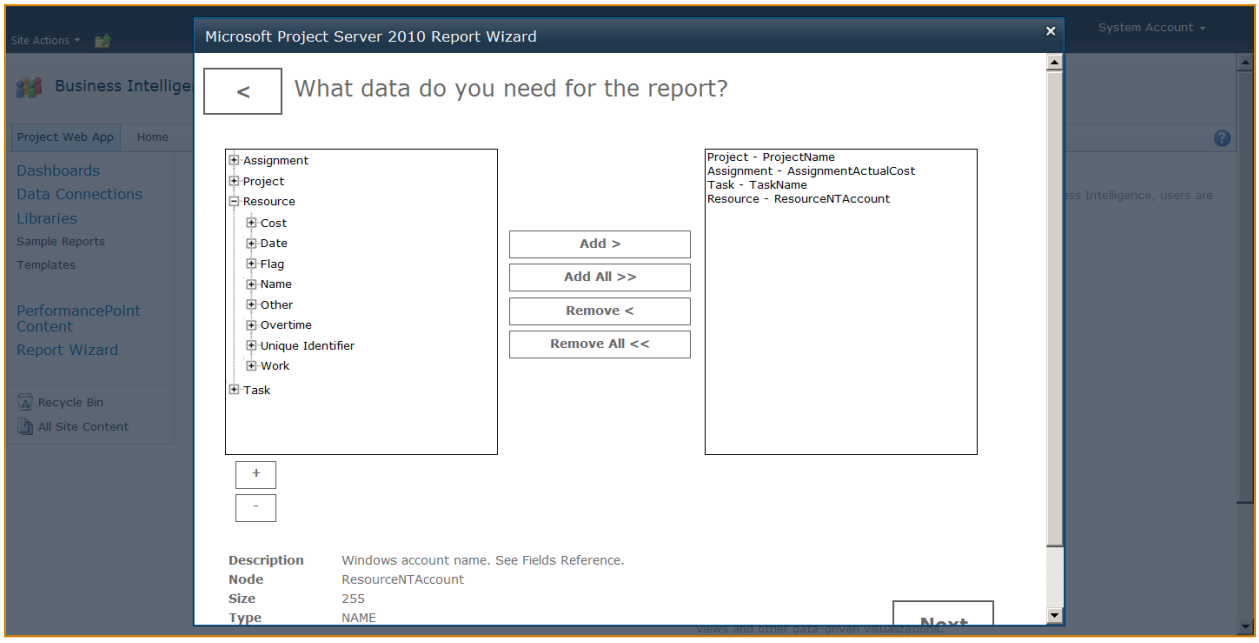
Modifying the ODC Field Settings

Note that some fields may create issues when added to ODC files. Specifically, the ODC will not display the unique IDs or multi-value fields without special instructions and coding.

Tips and Tricks: ODC Files

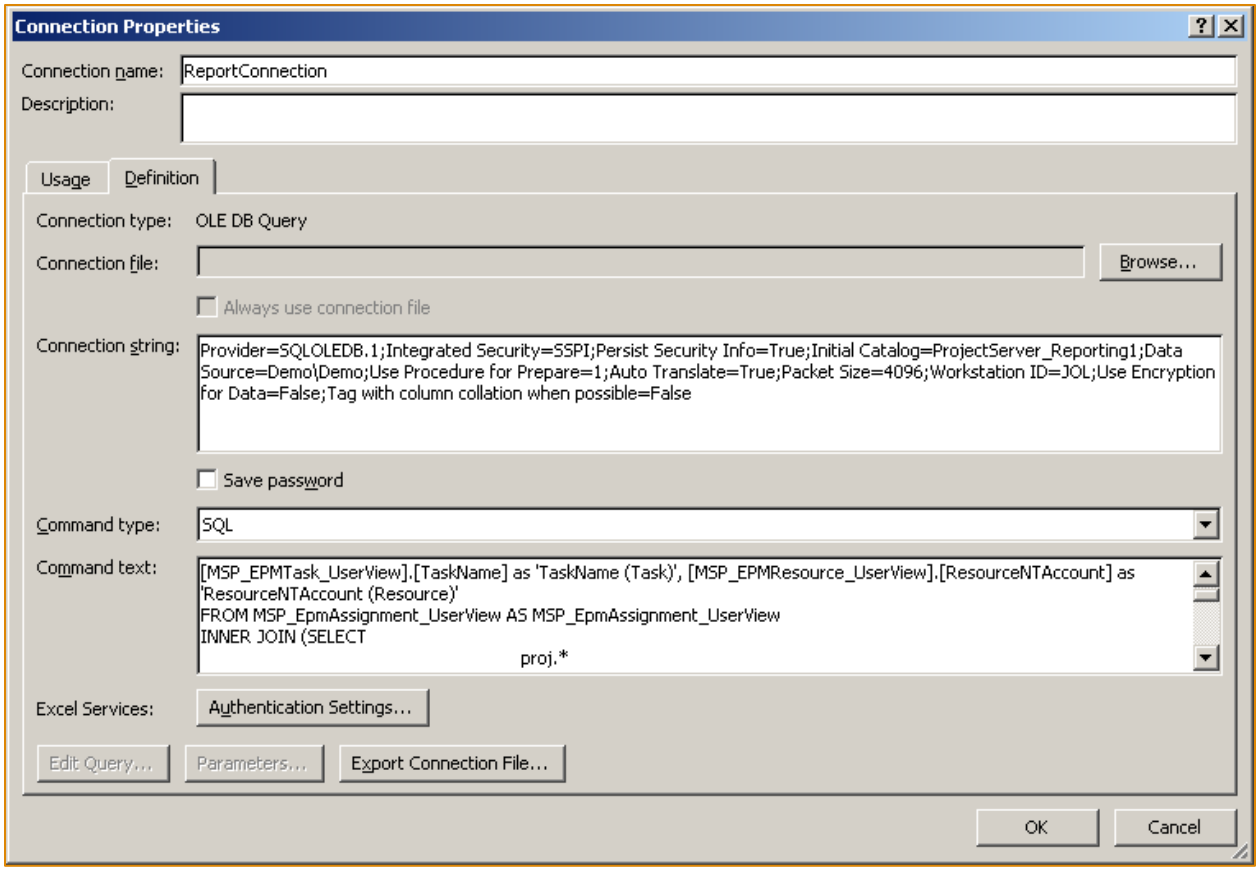
ODC files allow for the combination of multiple tables or views to create useful Excel and Visio reports. One trick to leverage this feature is to add the parent task to task summary reports.

For users who are not comfortable developing ODC scripts, Microsoft has provided a helpful tool that allows users to select the desired fields and then generates the required ODC file. The ODC Report Builder tool is a free download and is available as part of the Project Server 2010 Solution Starter pack (http://code.msdn.microsoft.com/P2010SolutionStarter).



Building an ODC with the Report Wizard

After you select the fields that you want, the Report Wizard generates the appropriate ODC file and posts it to the selected SharePoint library.



Generating an ODC file from the Report Wizard

Report Templates

Project Server 2010 is shipped with default report templates and Office Data Connections that can be used once the configuration is complete. These Excel-based report templates can either be used as is or can be used as a basis for creating additional reports.

There are also preconfigured blank templates provided that are connected to the included ODCs to help users create new reports quickly. For each OLAP database, a blank template and accompanying ODC is created for each cube within the OLAP database.

Each time you create a Project Web App site the following reports are available in the Business Intelligence Center:

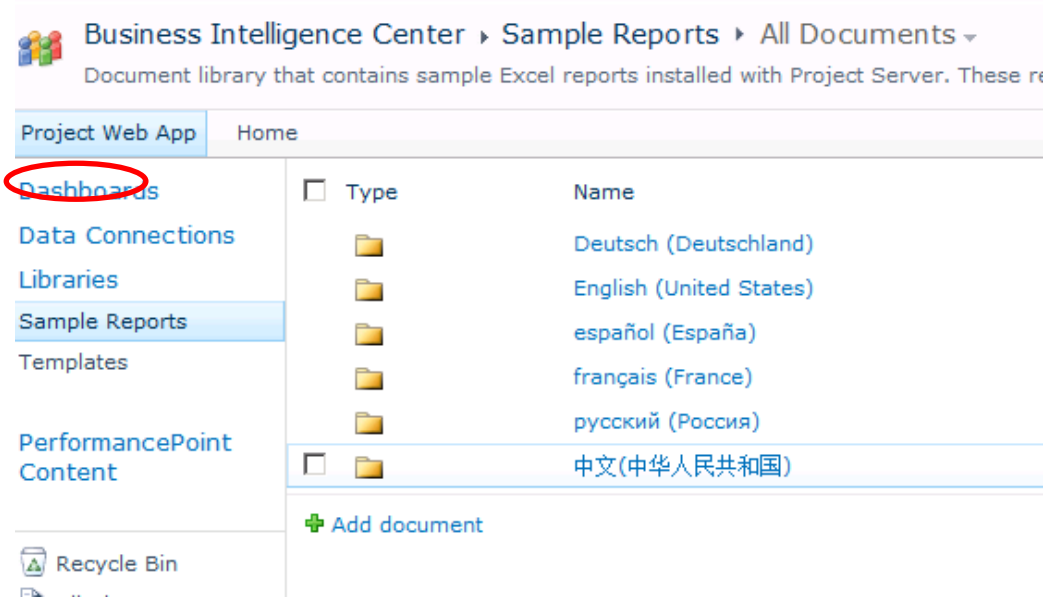
Here is the list of the default Templates:

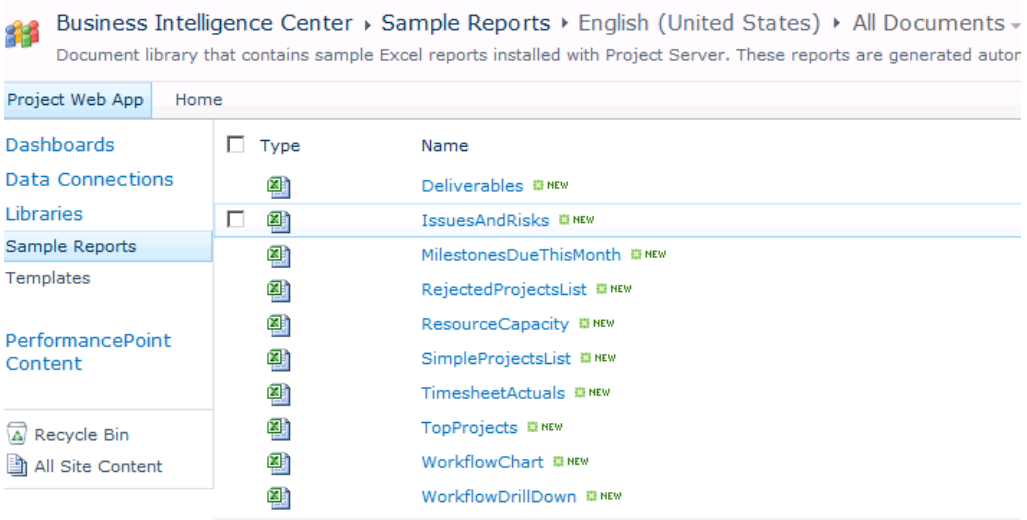
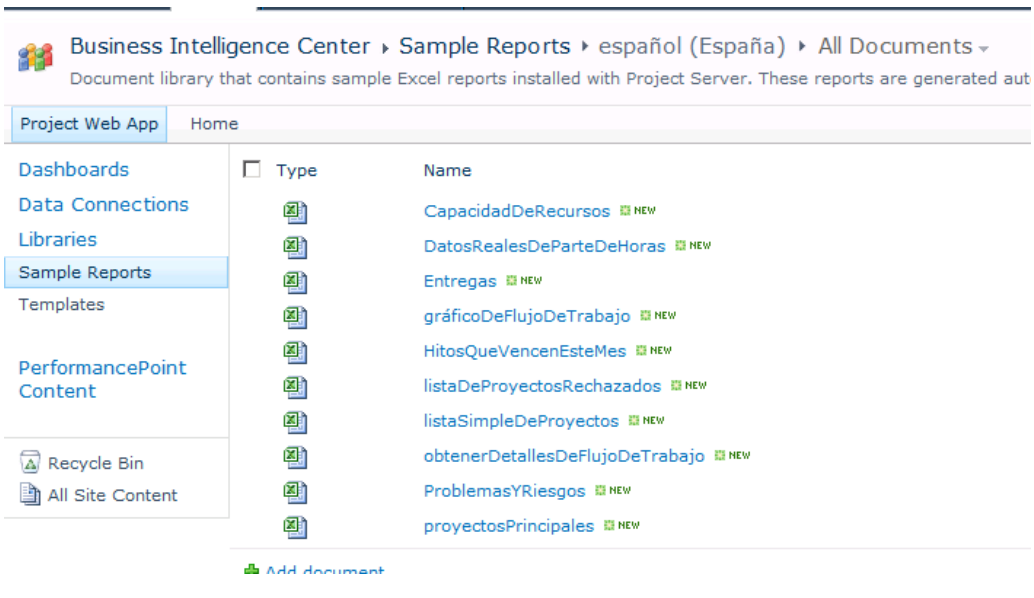
- Dependents Project (DependentProjects)
- Issues
- Project and Assignments (ProjectsAndAssignments)
- Resources
- Risks
- Timesheet

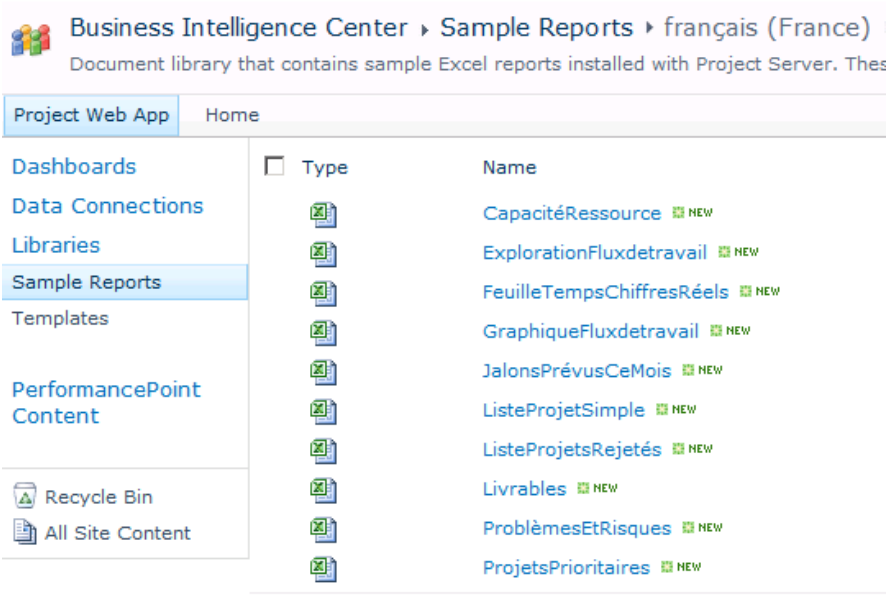
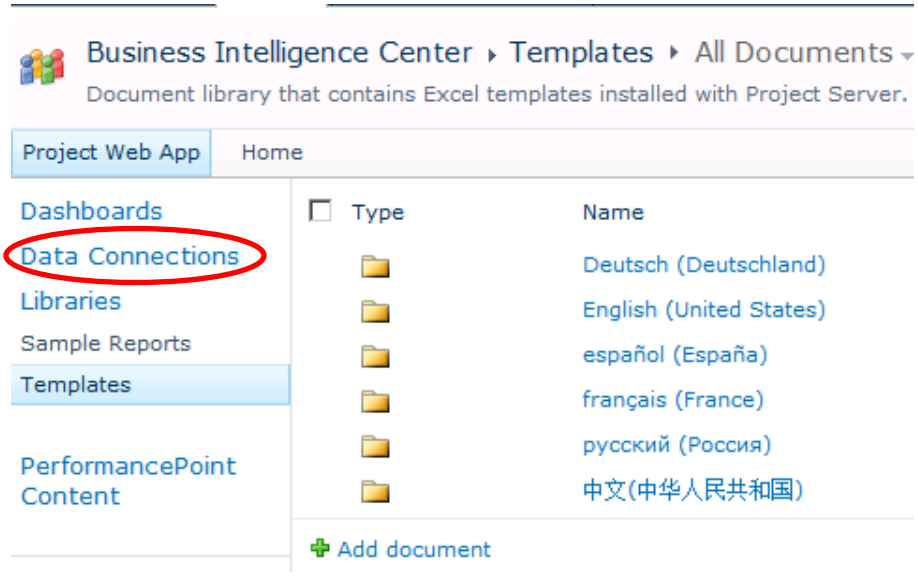
Sample Reports

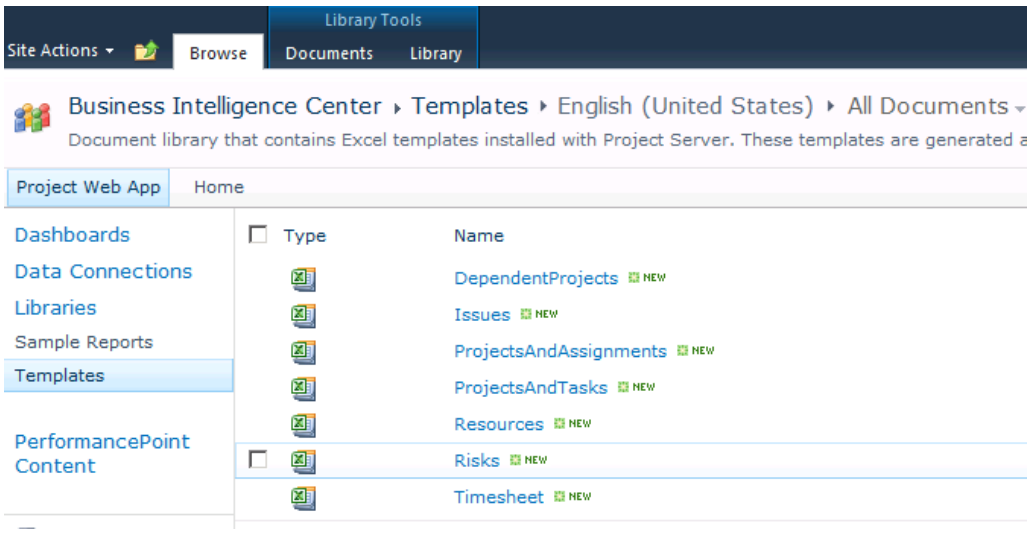
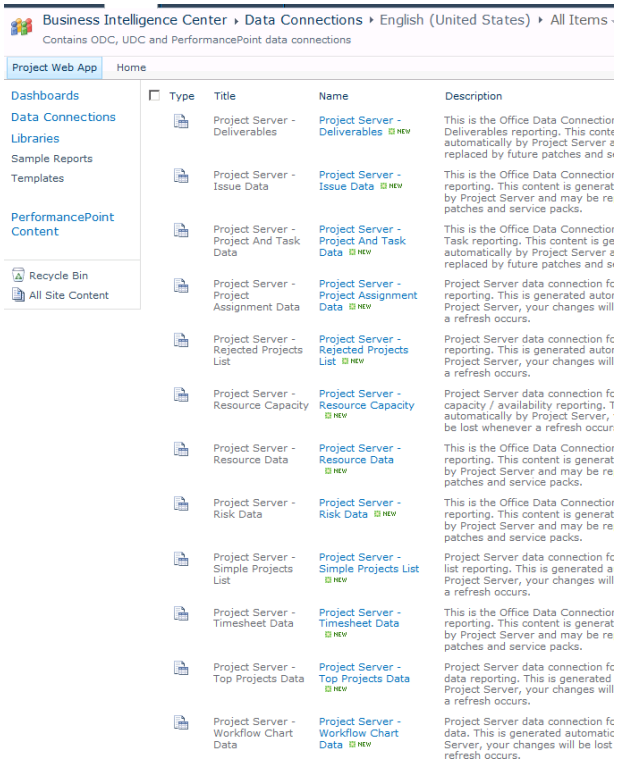
- Enterprise Project Management
    - Simple Project List (SimpleProjectList)
    - Milestone due This Month (MilestonesDueThisMonth)
    - Resource Capacity (ResourceCapacity)
- Timesheet
    - Timesheet Actuals (TimesheetActuals)
- SharePoint Lists
    - Deliverables
    - Issues And Risks (IssuesAndRisks)
- Portfolio
    - Rejected Project List (RejectedProjectList)
    - Top Projects (TopProjects)
    - Workflow Chart (WorkflowChart)
    - Workflow Drill Down (WorkflowDrillDown)

The following screen shots present how you can access these reports that come with the product:

Actions	Screen
Navigate to Project Web App Business Intelligence Center This is a SharePoint site that is a subsite of the PWA site.	
Under **Libraries** on the Quick Launch are links to two default document libraries that are created during provisioning of PWA. They are **Sample Reports** and **Templates** libraries. Here are the Sample Reports with some different language packs installed: • Deutsch • English • Español	

Actions	Screen
• Français • Russian • Japanese	
To access a report (once Excel Services is configured), drill into the language folder. For English (United States) You see the list of reports	
for Spanish	

Actions	Screen
And for French	
If you navigate to the Template folders Templates with the different language packs installed.	

Actions	Screen
For English (United States)	
Associated with each of these templates or sample reports there are predefined Office Data Connections (ODC) files	

Securing your reports

The security for Reporting is distinct from Project Web application security, because Report viewers may not be part of PWA Users. The Business Intelligence Center uses the native SharePoint Security.

The PWA Administrator administers the user with rights to manage the BI Center suite. By default all the PWA members have read-only access to the BI Center site.

There is a need to use a specific SQL Security Group for Project Server Report Author with DB_Reader access rights to the relational database. We recommend that you use an AD group to contain the list of users.

The following security model is used for Reporting:

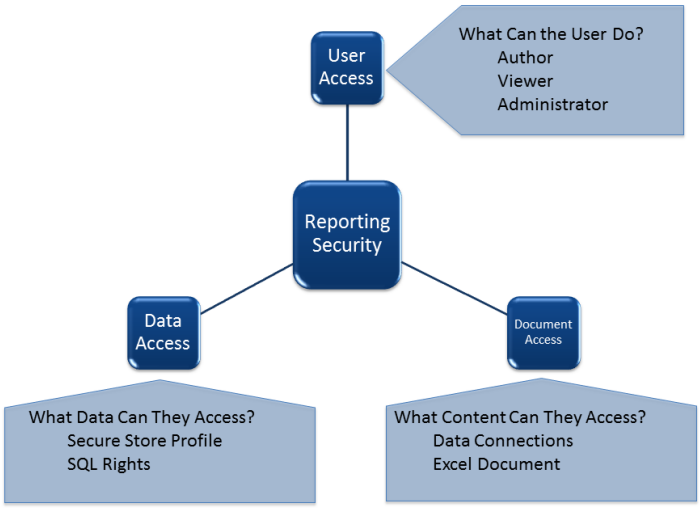


Figure 3 - Reporting Security Model

Excel Services Security Model

The security model for Excel Services is based on the concept to ensure:

- Data integrity
- Data quality

An administrator must be able to centrally manage shared resources and user access to corporate intellectual property contained in workbooks. To accomplish this, the following must be specified with Excel Services:

- **Trusted file locations**: These are SharePoint document libraries, UNC paths, or HTTP Web sites that have to be explicitly trusted before Excel Calculation Services is allowed to access them. Excel Calculation Services opens workbooks that are stored in trusted file locations only.
- **Trusted data providers**: These are external databases that Excel Calculation Services is explicitly configured to trust when processing data connections in workbooks. Excel Calculation Services attempts to process a data connection only if the connection is to a trusted data provider.
- **Trusted data connection libraries**: These are SharePoint document libraries that contain Office data connection (.odc) files. The .odc files are used to centrally manage connections to external data sources. Instead of allowing embedded connections to external data sources, Excel Calculation Services can be configured to require the use of .odc files for all data connections. The .odc files are stored in data connection libraries, and the data connection libraries have to be explicitly trusted before Excel Calculation Services will allow workbooks to access them.
- By default, cross-domain workbook and data connection access is not allowed.

List of the objects to configure/use in Project 2010

The following objects are configured and used in Project 2010 for Reporting:

- Custom Fields and Lookup Tables
- Departments
- Business Intelligence (BI) Center
- OLAP Database Management and Analysis Server Setup

The Business Intelligence Center is a SharePoint site that is a subsite of the PWA site. The URL will look like this: ***http://servername/pwa/ProjectBICenter***

List of the features to configure/use in SharePoint Server 2010/Excel Services

The following features are configured and used in Excel Services in Microsoft SharePoint Server 2010:

- Secure Store Service and service application

- PerformancePoint Services and service application
- Excel Services and service application
- Data Connections
  - Office Data Connection File (ODC)
  - Universal Data Connection File (UDC)

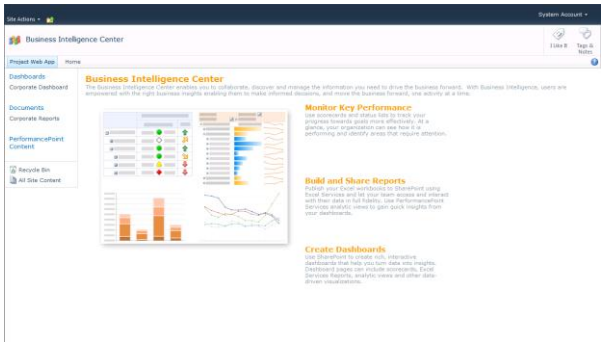
Blank data-connected ODC files are automatically provided with the product.

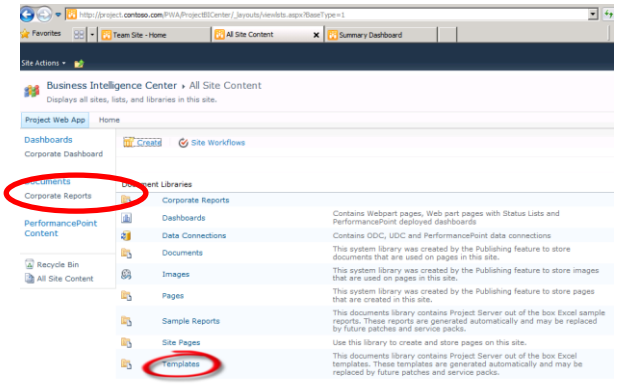
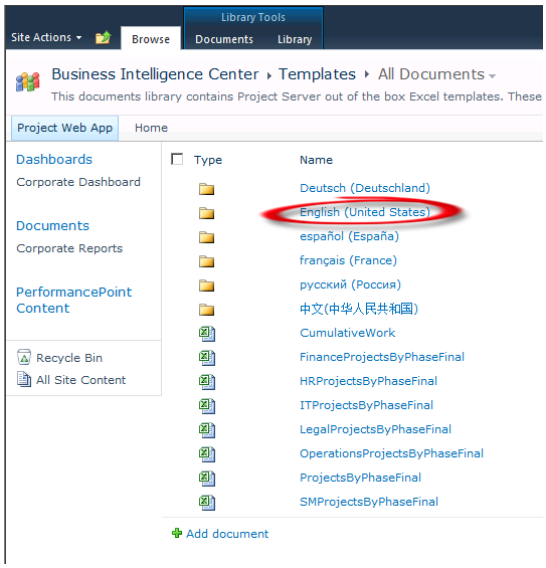
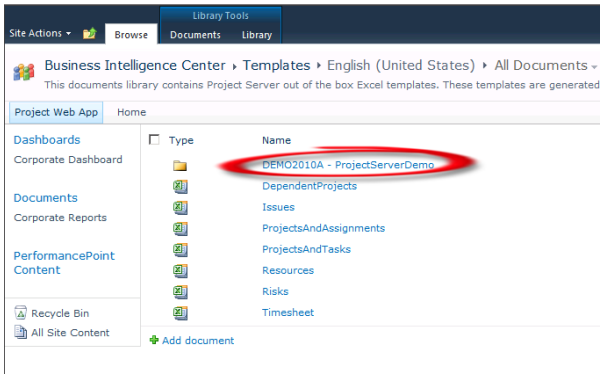
3. Build Your First Excel Report against OLAP Data in Ten Minutes

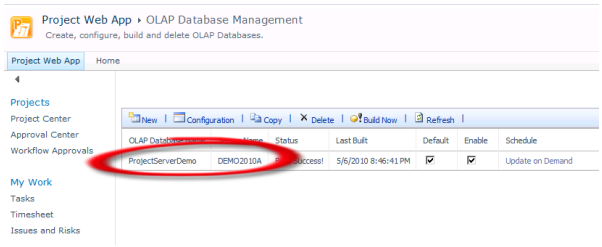
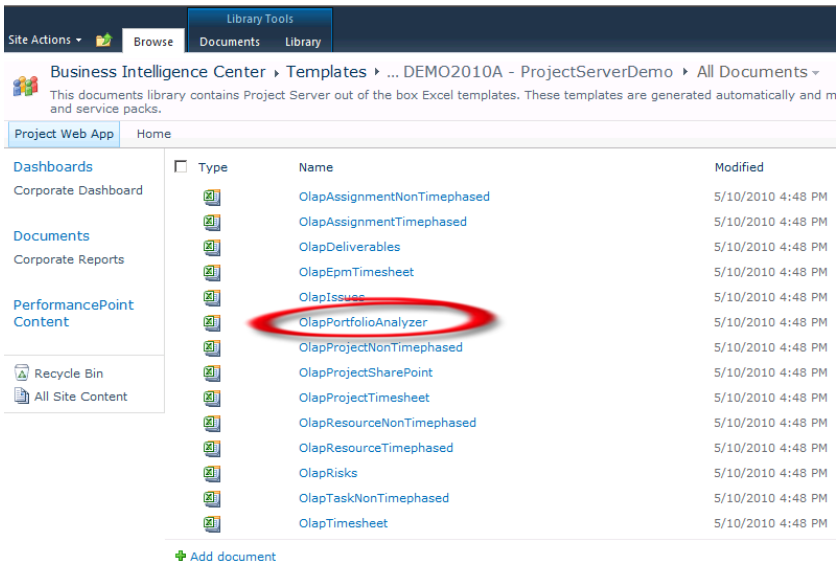
Introduction

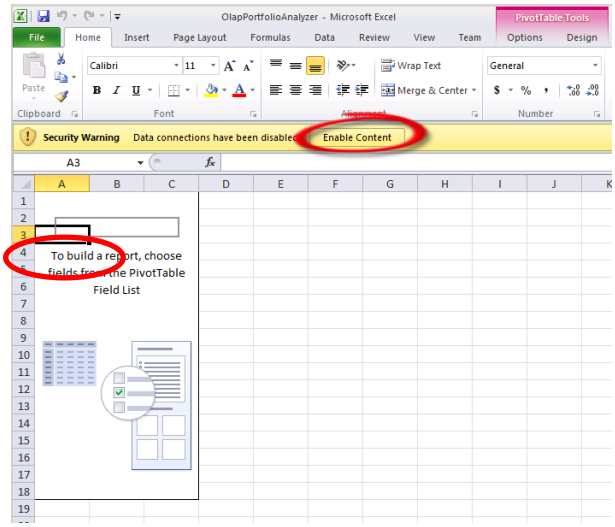
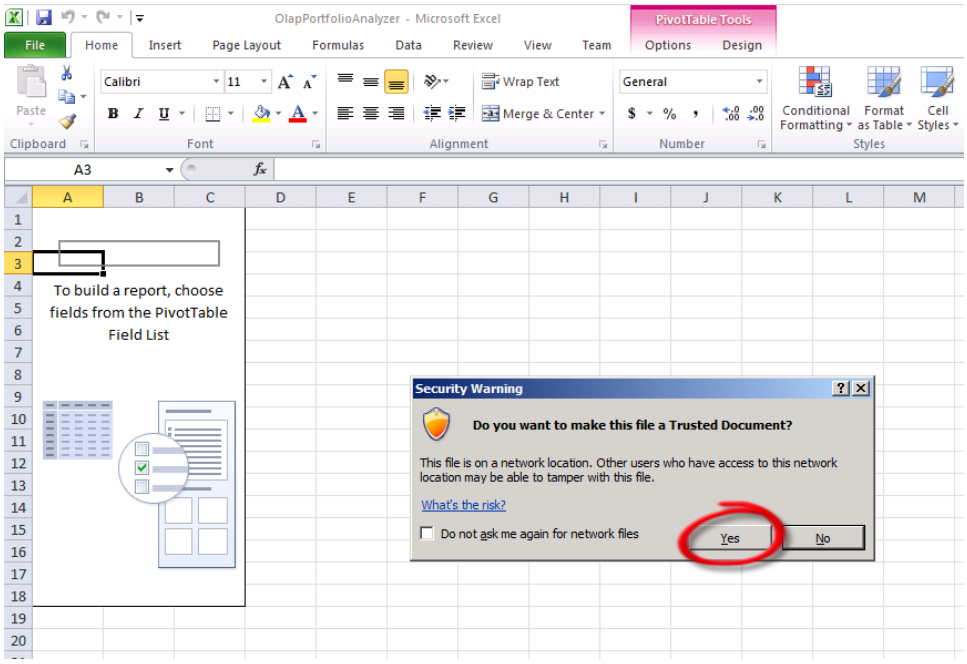
In this chapter we will build our first report with Excel and Excel Services to analyze the capacity and planned work for each department over time.

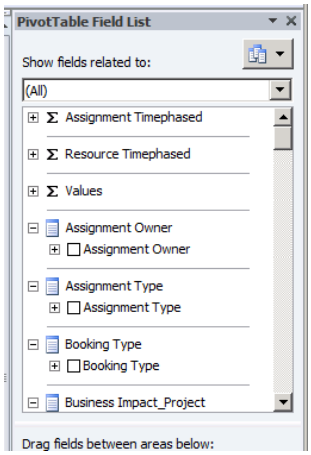
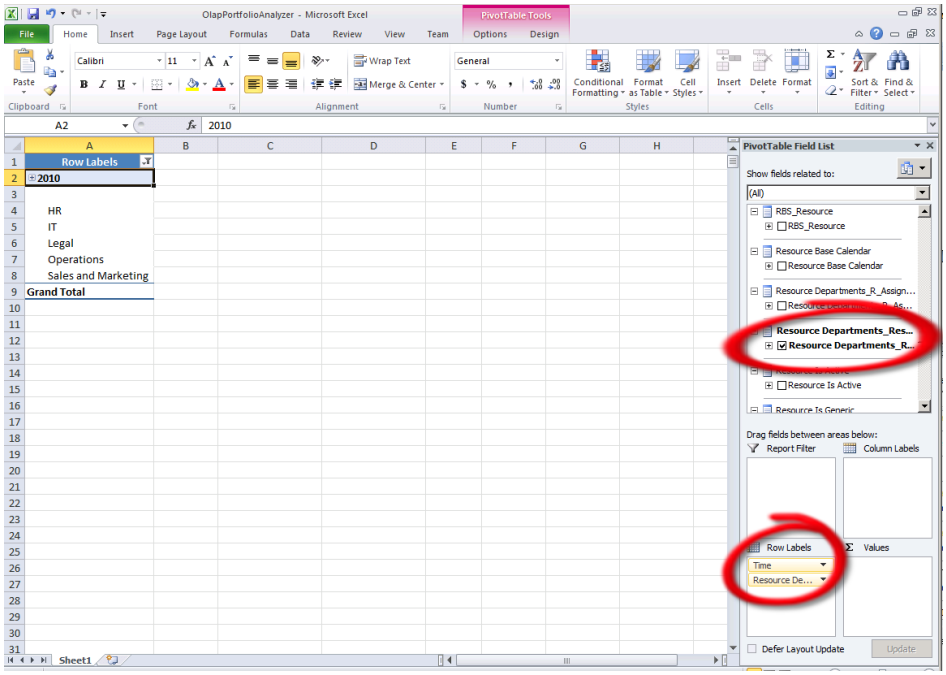
The following screen shots present how you can create this first report:

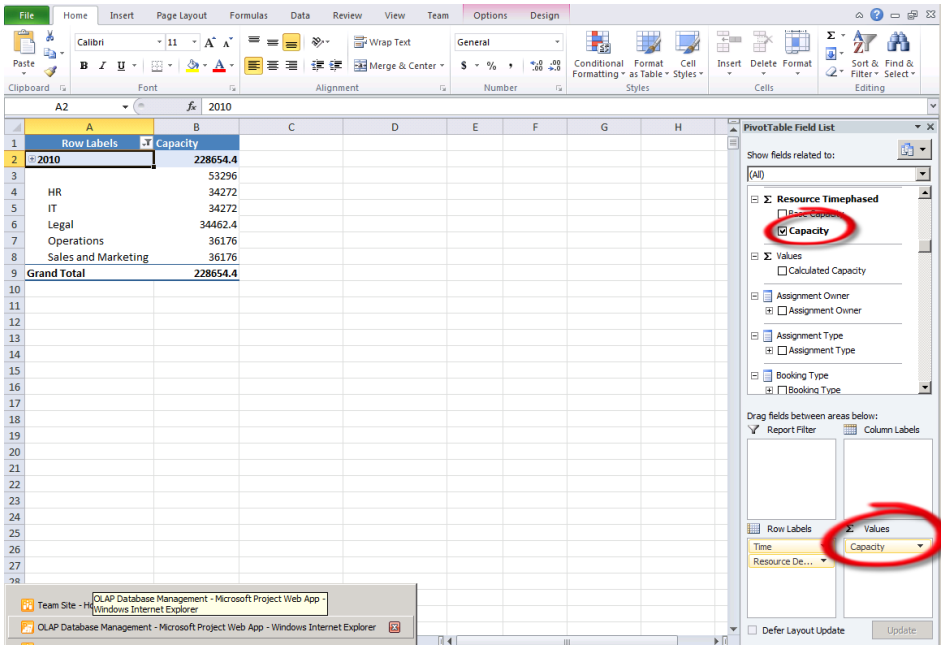
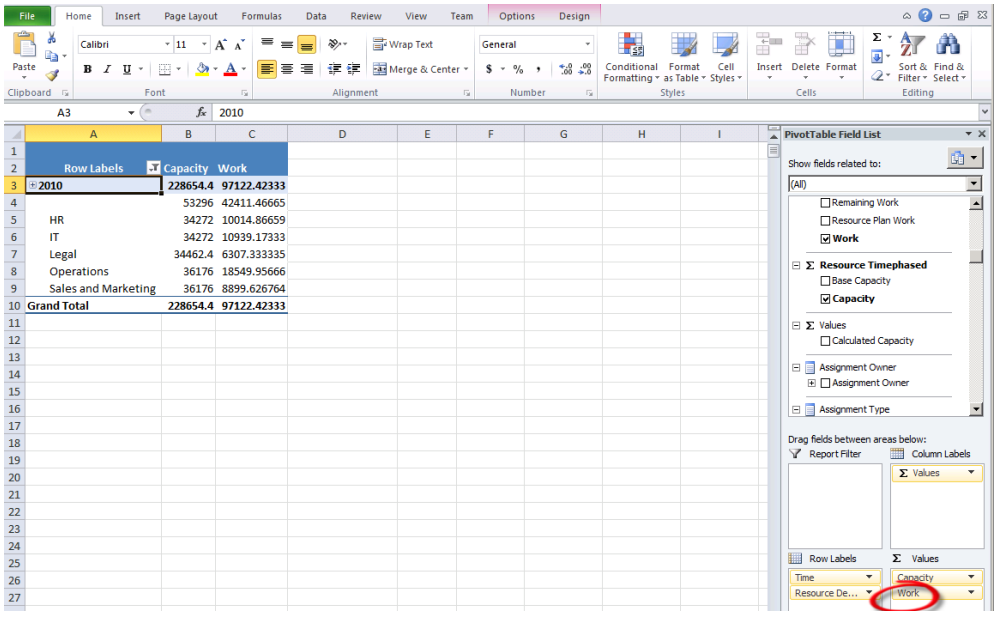
Actions	Screen
Navigate to Project Web App Business Intelligence Center This is a SharePoint site that is a subsite of the PWA site. Select **Documents** in the Quick Launch menu.	

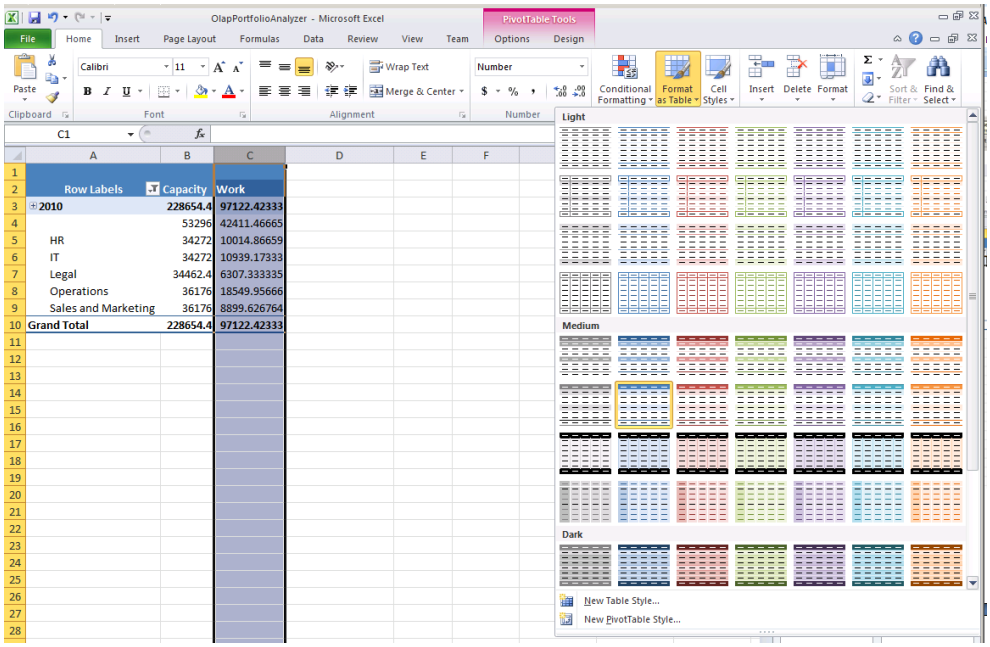
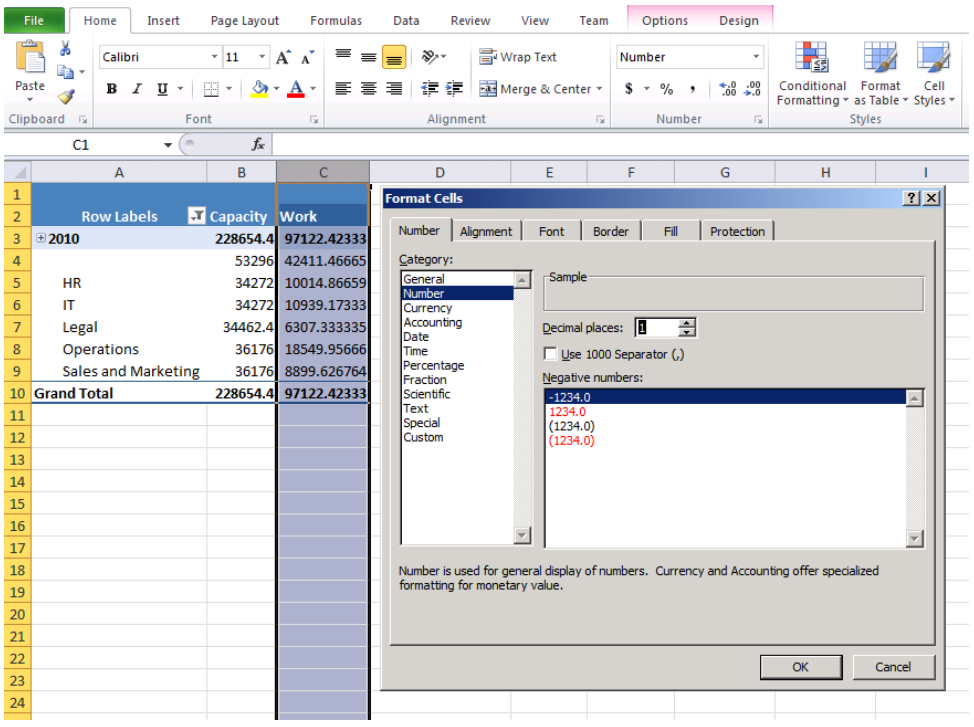
In the list of the Documents libraries, select **Templates** to access the list of report templates	
In the list of templates, select the folder that matches your language, in this case **English.**	
In the list of templates for the English language, select the folder that matches the SQL Server Analysis Services server name and the name of the OLAP database created when building the cubes in Project Web	

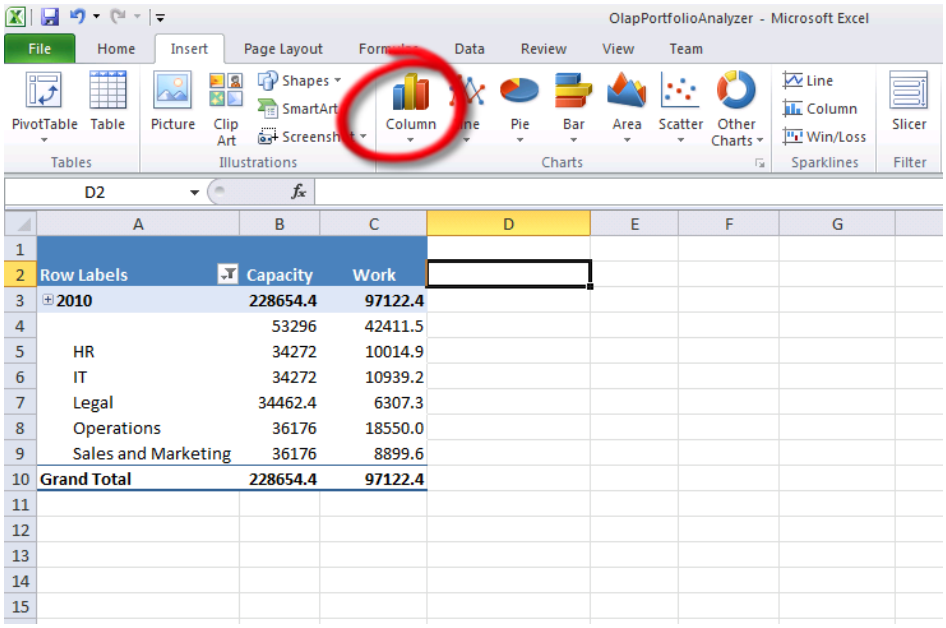
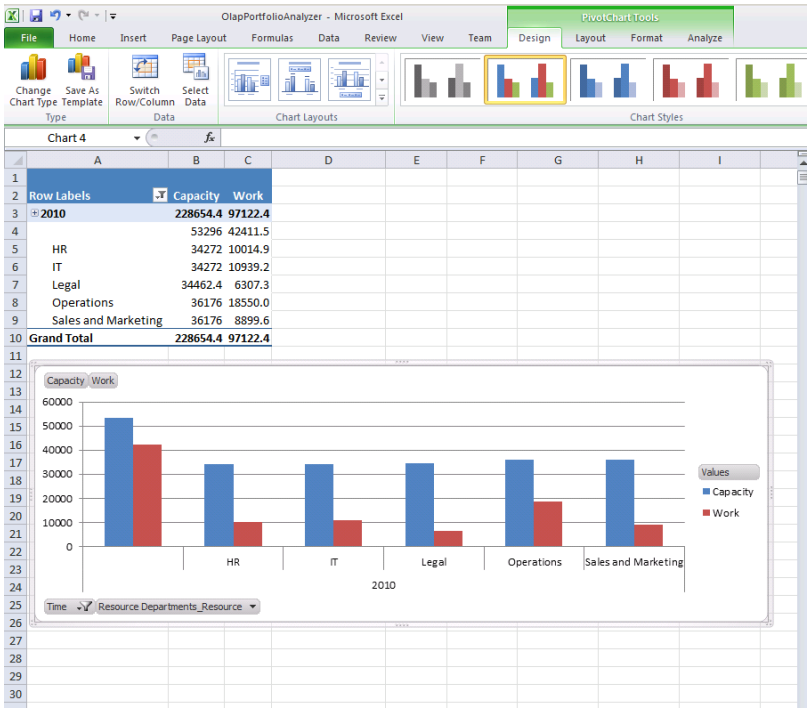
App	
The SQL Server Analysis Services server name and the name of the OLAP database are entered when building the cubes in Project Web App	
Select the template '**OLAPPortfolioAnalyzer**' from the list of the 14 Excel report templates which corresponds to the 14 cubes created when the cube building process is run in PWA. The OLAPPortfolioAnalyzer template reads the data from the Portfolio Analyzer cube that contains all dimensions and measures as it was in the previous versions of Project Server since	

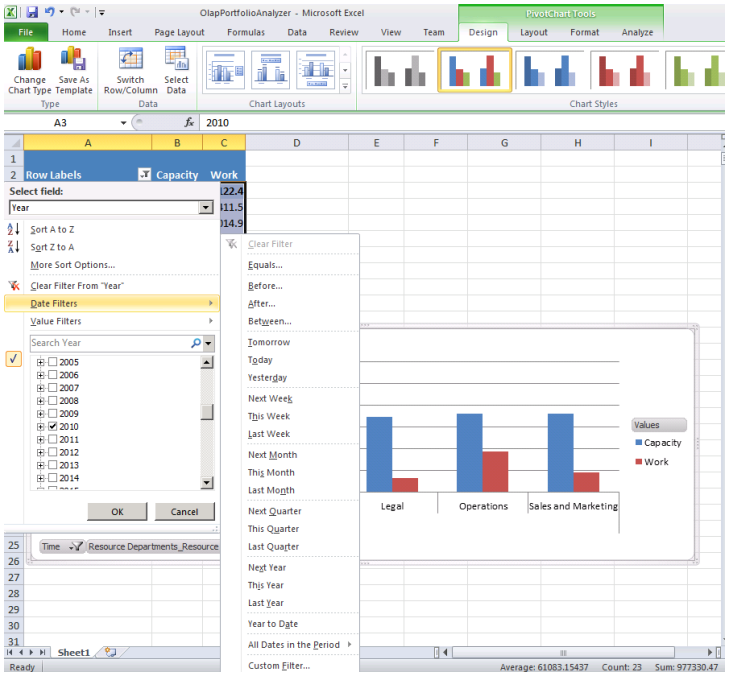
2002.	
When you select the template, the Excel client starts and loads the template. By default, the data connections are disabled and must be explicitly allowed with the 'Enable Content' button. Once that is enabled, the list of fields available will be displayed.	
Answer 'Yes' to the prompt.	

The list of fields from the cube 'OLAPPortfolioAnalyzer' is displayed. There are some Time-phased measures at the assignment level as well as at the Resource level. In our case, Capacity is at the resource level and Work is at the Assignment level.	
We start by selecting the Time dimension and placing it in the Row Labels. Then we select the Resource Departments dimension and place it below the Time in the Row Labels as well.	

Then, we select the first measure ‘Resource Capacity’.	
Then we add the second measure, ‘Work’ from the Assignment Time-phased measures.	

The rich menu bar can be used to format the style of the cells.	
The rich menu bar can also be used to format the format of the cells. In this case, we want to only show one decimal for the work.	

Once we have completed the PivotTable, we can also insert a PivotChart representation of the same data set shown in the PivotTable. Select Insert from the menu bar and choose the PivotChart style.	
The final result is shown across.	

It is also possible to set Date Filters to restrict the period of analysis to a specific date range.	
Once the report is completed, select File and Save As to save the new report to a report library and avoid overwriting the report template. There are two ways to save the new report. Save As locally on a desktop folder and then upload the new report to a trusted Excel Services report location, or Save As/Publish directly to the SharePoint Excel Services Report library.	

4. Authoring Reports against Relational Data

Here are the different steps when authoring a Report:

- Basic authoring
- Report formatting
- Filter condition
- Pivot Charts
- Publishing

It is important to name the different objects in your Excel workbook (sheet, filter parameter, chart), so that they can be published on the web and rendered by Excel Services or PerformancePoint Services in dashboards.

Tips & Tricks

Do not modify default names

In the BI Center, we recommend NOT to rename or delete the default content or its containing folders, to avoid maintenance problems when service packs are installed.

You can create your own folders with a specific naming convention (that includes your company name, for example Contoso Reports), so that you avoid name clash with future releases.

Microsoft reserves the rights to update default content when patches or services packs are released.

Do not modify default ODC

The ODCs that are provided by Microsoft may be patched by subsequent Service Packs and Patches. Therefore, instead of modifying the predefined content, we recommend that a copy be made and

modified. Another alternative is to create a trusted Data Connection Library folder where all the custom ODC files shall be stored.

How you would create an Excel Report the old way

Here is a fairly exhaustive list of all the steps that would be required to build and present an Excel report:

- Create a Report Center Site
- Create an Office Data Connection (ODC)
- Set PivotTable to sort fields in Data Source Order
- Link PivotTable to ODC
- Get the Server Name
- Create an Excel Template
- Set ODC to recognize language packs
- Set up the Unattended Service Account
- Configure Excel Services
- Set up trusted folders
- Create PivotTable
- Create a SQL query to pull required data
- Get the Database Names
- Set Office Data Connection to auto refresh on open

Thanks to the new features that come with Project Server 2010, almost all these steps are done for you automatically:

- Included reports for dashboards and as report starters
- Pre-connected templates for quick report creation
- Automatic Template creation
- Business Intelligence Center

Creating a New Excel Report for Project Server 2010 with Excel 2007-2010

In this chapter we describe the steps involved to create the reports by using Excel 2010.

- Build the report
- Reuse an existing report

- Publish the report

Excel Services reports function very much like Excel workbooks and can appear similarly when you view them, whether you view an Excel workbook on your computer or in a SharePoint document library.

Excel Services reports can include a variety of standard Excel features and functionality:

- Conditional formatting
- Formulas
- Charts

When you publish an Excel workbook to Excel Services, your workbook becomes the data source for an Excel Services report type in the Dashboard Designer.

Creating any type of Excel Services report is a two-step process:

1. You begin by using the wizard to create the basic report structure.
2. Then you can select the data to display in the report.

External Office Data connections (.ODC Files)

External Office Data Connections, a component of Excel Services, is used to connect to external data sources.

Excel Services processes external data connection information that contains everything the server needs to connect to a data source, including

- How to authenticate
- which connection string to use
- Which query string to use
- Application ID (used with secure store)
- Where and how to gather credentials to use for the connection

These connections can be defined in two places:

- Embedded within workbooks
- in .odc files

The connection information is identical in both places. The .odc files are small files that persist connection information in plain text and in a format that is reusable.

You can use the Excel 2010 client to author and edit .odc files and connections embedded in workbooks. In the Excel 2010 client, you can run the Data Connection Wizard or configure the settings in the Connections properties page. You can also export an .odc file based on these settings. The Connections properties page shows connection information, including Excel Services authentication properties.

Workbooks can contain links to .odc files and embedded connection information. This enables workbooks to retrieve the .odc file, read the contents, and attempt to connect to an external data source if the embedded connection information fails. The .odc files must be managed and maintained to ensure that they contain accurate data connection information

The ODC are stored in the Data Connection library in the BI Center. By default, the library uses the SharePoint Server 2010 Content approval workflow to control visibility to an ODC. This provides change control on this shared reporting resource.

Creating/Updating a new ODC

If any new ODC is added to the library or changes are made to an existing ODC, it will appear in a pending state. While pending, only the person who last changed the file can use it. A user with workflow approval rights must go into the library and approve the new/changed ODC to make it visible to all.

ODCs that are provided by Microsoft may be patched by subsequent service packs and pPatches. Therefore, instead of modifying the predefined content, we recommend that a copy be made and modified.

For the mechanics of how to create and modify Office Data Connections, please refer to this Office Online article:

http://office.microsoft.com/en-us/excel-help/create-edit-and-manage-connections-to-external-data-HA010167227.aspx

For an example of how to use Office Data Connections, refer to this Excel blog post:

http://blogs.msdn.com/excel/archive/2008/10/15/using-office-data-connection-files-odc-and-the-dataconnections-web-part-in-sharepoint-to-specify-external-data-connections-in-newly-created-excel-workbooks.aspx

Updating an existing Excel Report

In this chapter we describe how to add a new custom field to an existing report.

A new custom field has been added to Project Web App.

You want to add this new custom field to an existing report.

Configuring the Data in Project Server 2010

In this section we describe the precise steps required to add the custom fields of our example by using the administrative pages of Project Web App.

For more information about Custom Fields management in Project Server 2010, read the following resources:

http://technet.microsoft.com/en-us/library/gg663916.aspx

http://technet.microsoft.com/en-us/library/gg709725.aspx

In this example, we add a Project Custom Field that gives the category status for our Projects. This custom field uses a lookup table that gives the different status values: Internal Project, External Project.

We will see how these new fields can be made available in our existing reports.

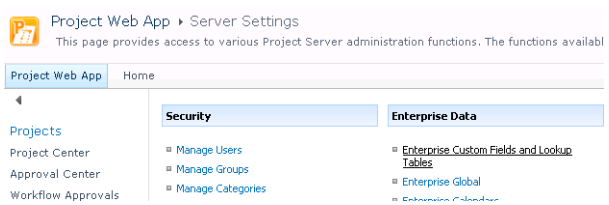
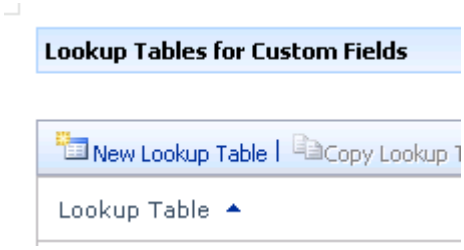
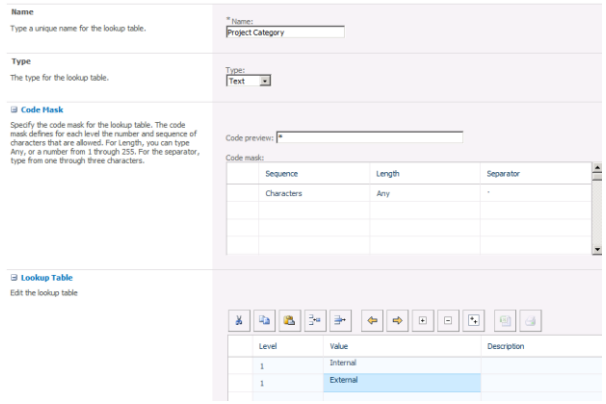
Custom Fields and Lookup tables

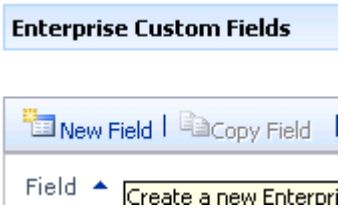
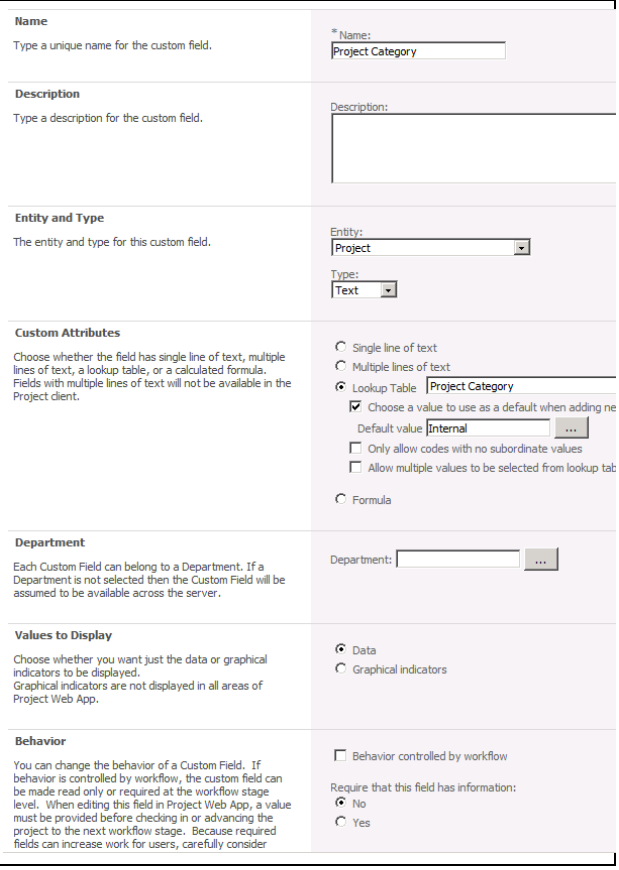
List of custom Fields used

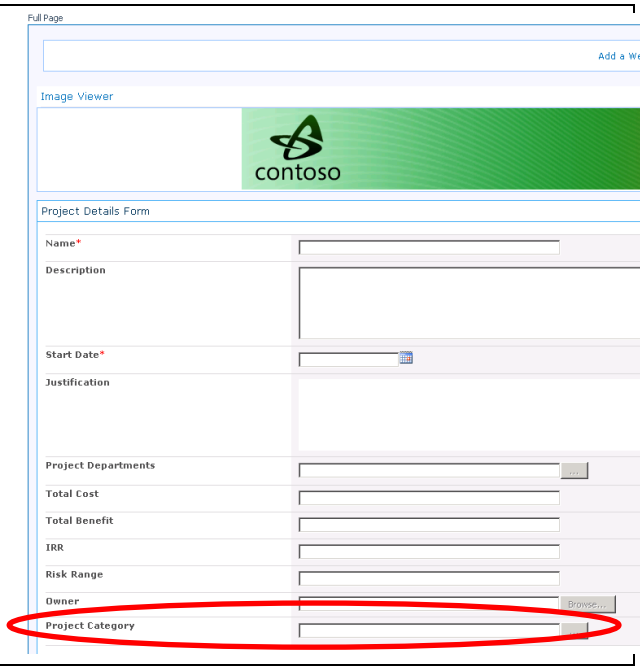
In this section we create first the Lookup tables (LT) and then the Custom Fields for our example.

Field name	**Custom Field (CF) or Lookup Table (LT)**	**Type**
Project Category	LT	Text: Value Internal/External

Project Category	CF	Project of Type Text Using a lookup table

Action	Page
In Project Web application, After selecting Settings/Server Settings: Select Enterprise Custom Fields and Lookup Tables	
Select the “New Lookup Tables” button	
Name: Project Category Type: Text Code Mask: * Lookup Table values: Internal External	

To create the Field “Project Category” Select the button “New Field” in the Enterprise Custom Fields section. Select Field button	
Name: Project Category Entity and Type: Project Text Custom Attributes: Lookup Table: Project Category Default value: Internal Behavior: not selected	

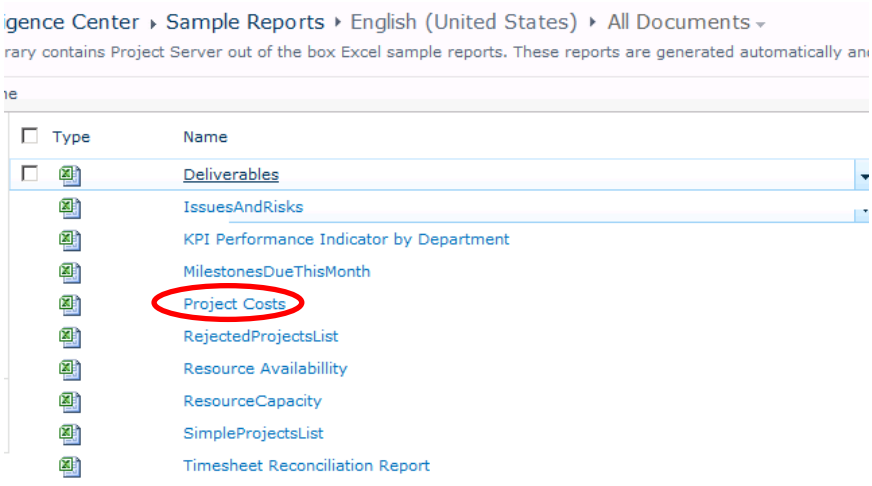
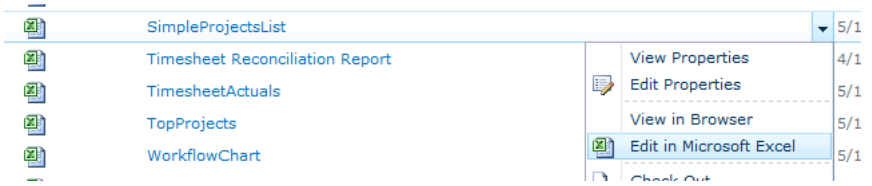
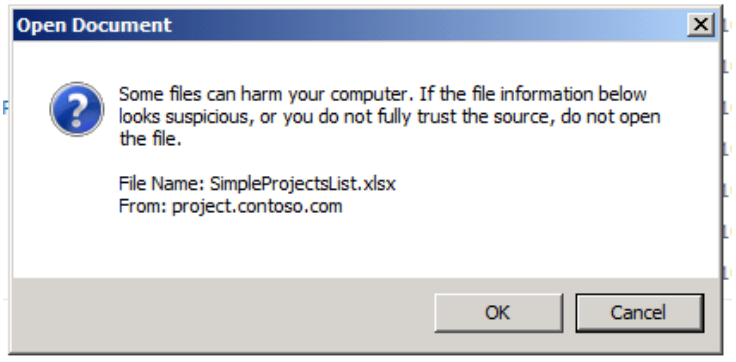
Note: You may want to add this field on a PDP (Project Detail Page) so that it can be updated. The detailed steps are not described here; see the white paper on Demand Management. Here is an example of the updated Project Detail Page with the new field	

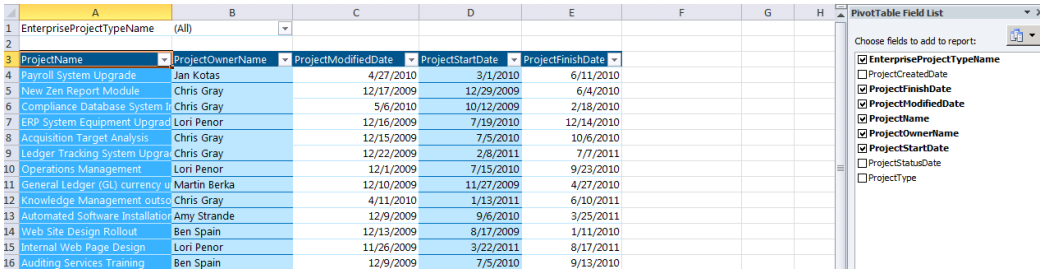
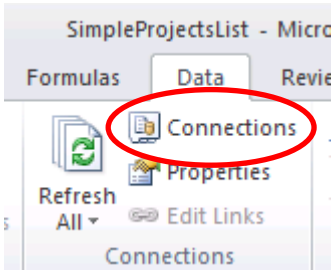
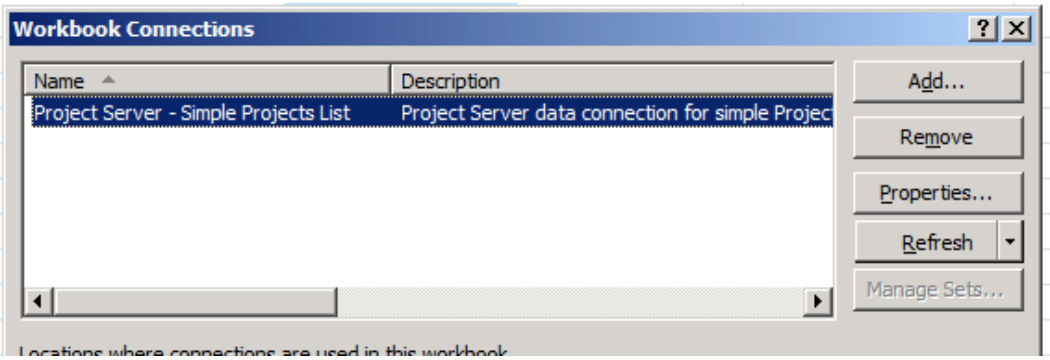
Creating a New Excel Report with Project Server 2010 Relational Data

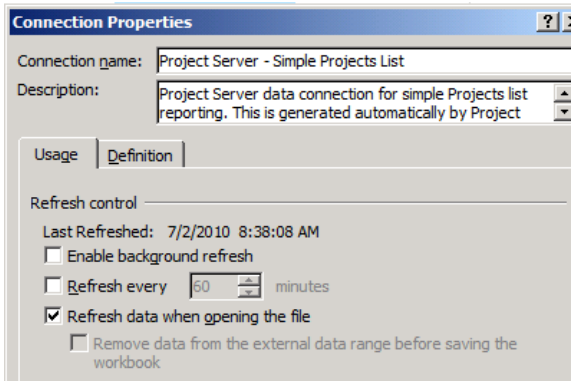
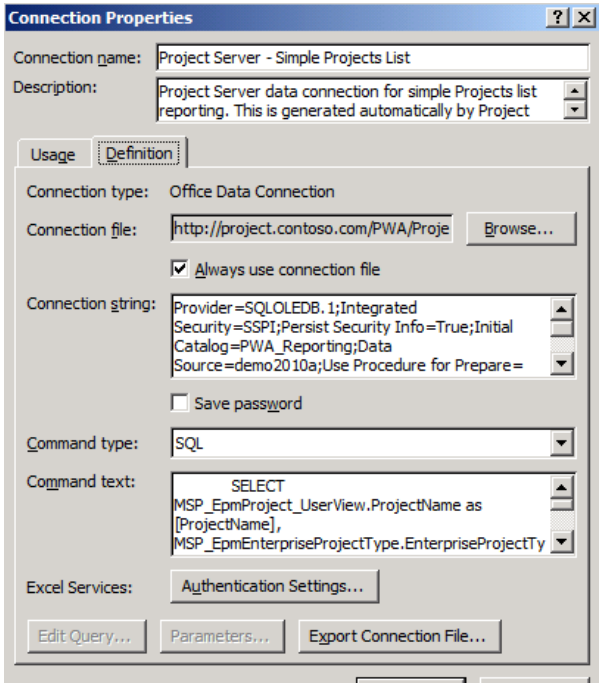
In this section we describe the precise steps involved to create the reports of our example using Excel 2010.

- We will reuse an existing report: SimpleProjectList
- Publish this report
- Show the result

Actions	Screen

Navigate to the BI Center and select the sample report: SimpleProjectList	
Select the report and use the contextual menu Edit in Microsoft Excel	
Depending on you security configuration, you may have a warning message that you can ignore:	

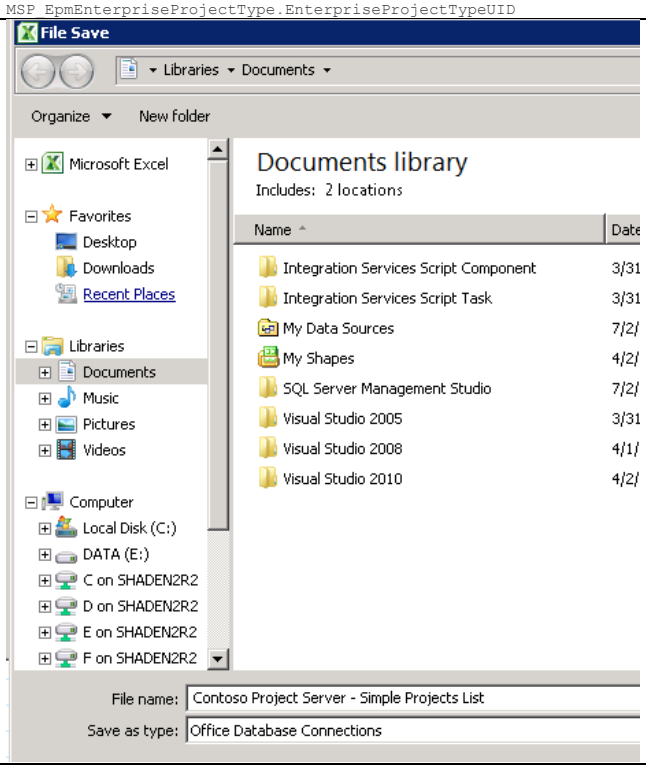
Excel 2010 is opened, the report is refreshed	
In order to add a field, the ODC file must be updated. Select the Data Tab and click the Connections button	
Select the existing connection and click the Properties button	

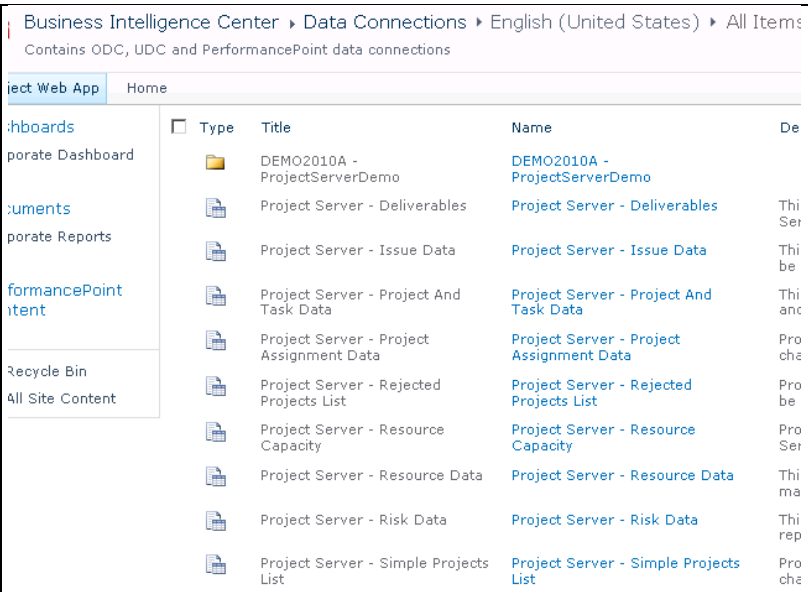
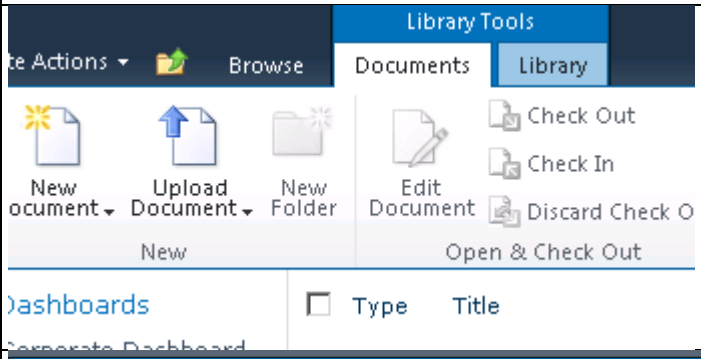
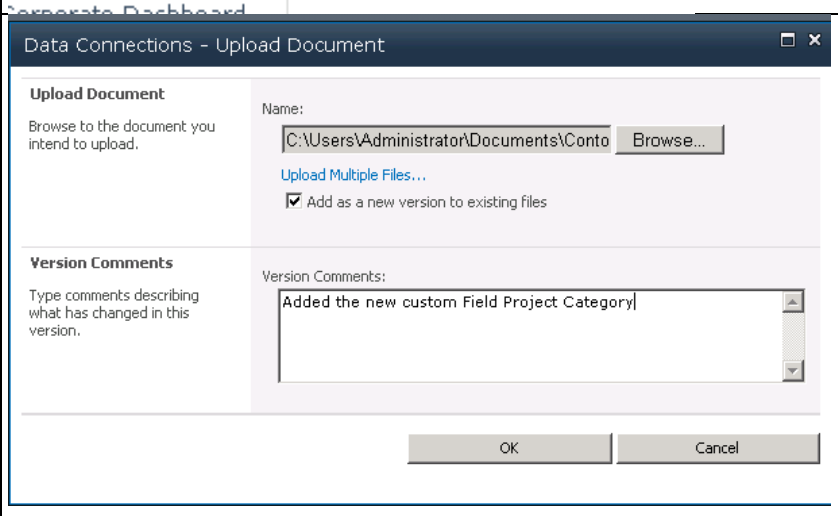
The Connection Properties dialog box appears. Select the Definition tab	
You can see the SQL query that must be updated	

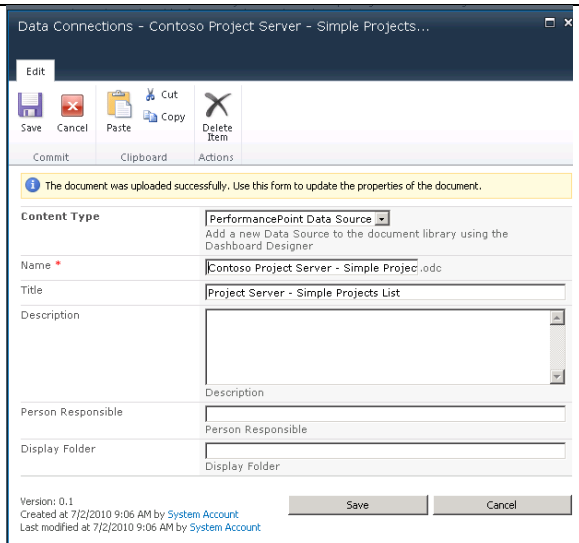
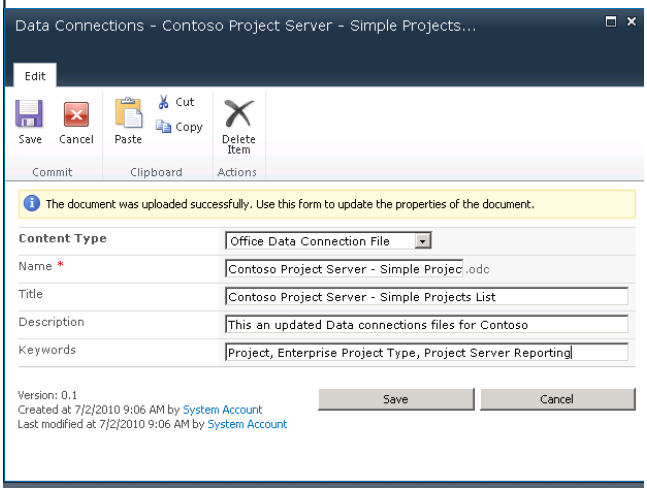
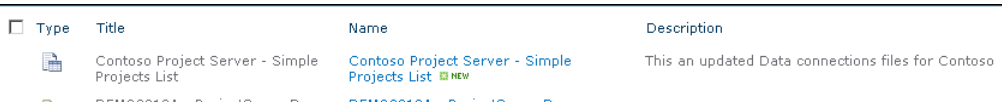
To update the SQL query, you can use SQL Server Management Studio or another SQL query tool.
 	
This is the original query.
	`SELECT`
	`MSP_EpmProject_UserView.ProjectName as [ProjectName],`
	`MSP_EpmEnterpriseProjectType.EnterpriseProjectTypeName as [EnterpriseProjectTypeName],`
	`MSP_EpmProject_UserView.ProjectOwnerName as [ProjectOwnerName],`
	`MSP_EpmProject_UserView.ProjectType as [ProjectType],`
	`MSP_EpmProject_UserView.ProjectStartDate as [ProjectStartDate],`
	`MSP_EpmProject_UserView.ProjectStatusDate as [ProjectStatusDate],`
	`MSP_EpmProject_UserView.ProjectFinishDate as [ProjectFinishDate],`
	`MSP_EpmProject_UserView.ProjectCreatedDate as [ProjectCreatedDate],`
	`MSP_EpmProject_UserView.ProjectModifiedDate as [ProjectModifiedDate]`
	`FROM dbo.MSP_EpmProject_UserView`
	`INNER JOIN dbo.MSP_EpmEnterpriseProjectType ON`
	`MSP_EpmProject_UserView.EnterpriseProjectTypeUID =`
	`MSP_EpmEnterpriseProjectType.EnterpriseProjectTypeUID`

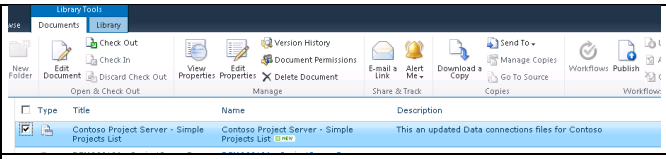
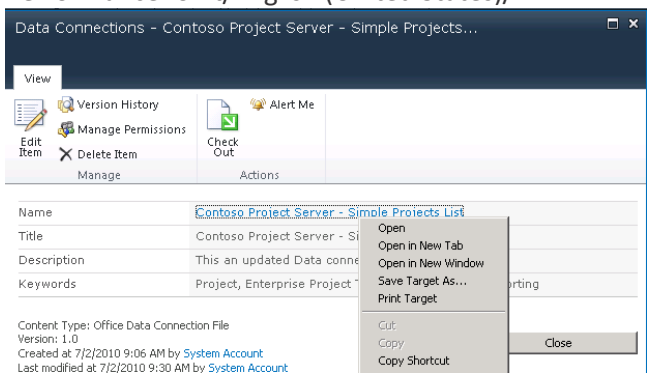
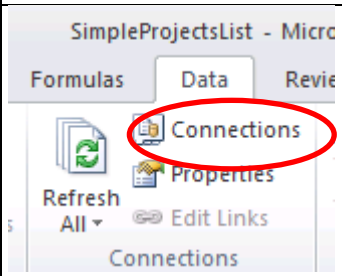
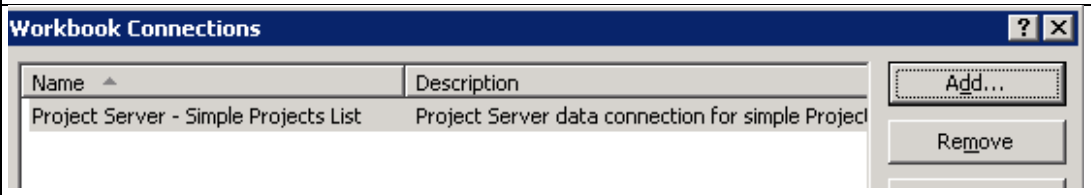
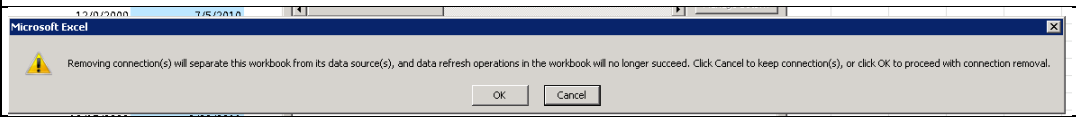
You need to add the column [Project Category] in the select statement.	
Export the ODC file to a local file and rename it: Contoso Project Server Simple Project List	
Close Excel without saving the report	

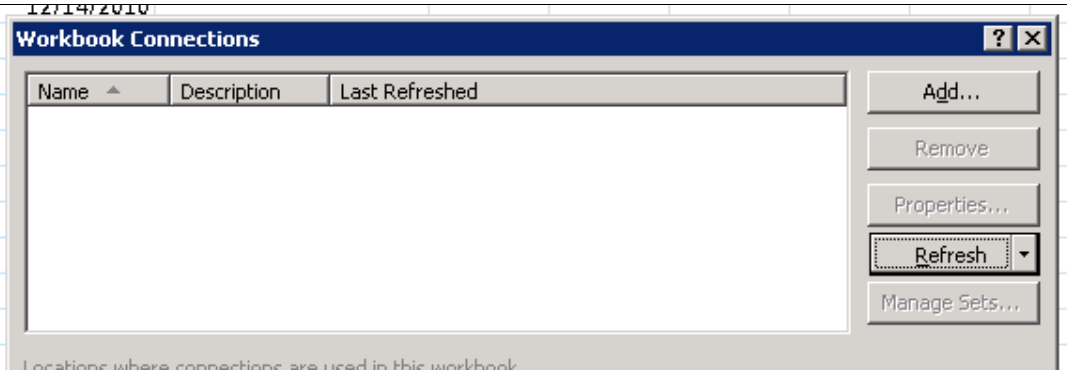
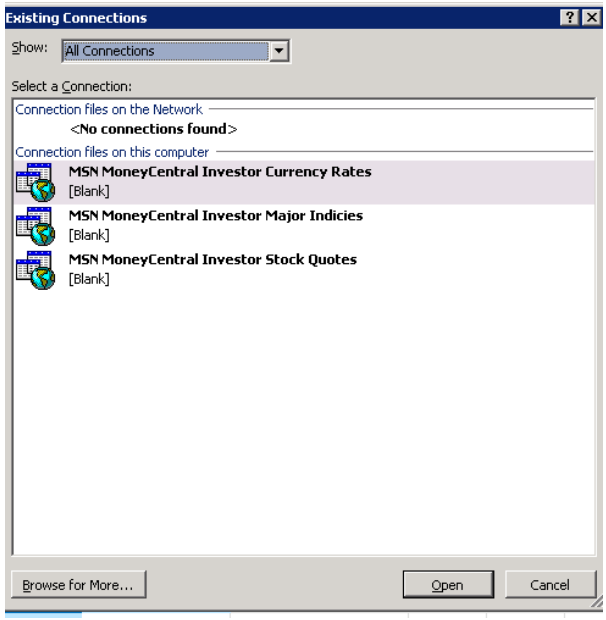
```
SELECT
                    MSP_EpmProject_UserView.ProjectName as [ProjectName],
          MSP_EpmEnterpriseProjectType.EnterpriseProjectTypeName as [EnterpriseProjectTypeName],
          MSP_EpmProject_UserView.ProjectOwnerName as [ProjectOwnerName],
          MSP_EpmProject_UserView.ProjectType as [ProjectType],
          MSP_EpmProject_UserView.ProjectStartDate as [ProjectStartDate],
          MSP_EpmProject_UserView.ProjectStatusDate as [ProjectStatusDate],
          MSP_EpmProject_UserView.ProjectFinishDate as [ProjectFinishDate],
          MSP_EpmProject_UserView.ProjectCreatedDate as [ProjectCreatedDate],
          MSP_EpmProject_UserView.ProjectModifiedDate as [ProjectModifiedDate],
          MSP_EpmProject_UserView.[Project Category]   as   [Project Category]
FROM      dbo.MSP_EpmProject_UserView
INNER JOIN dbo.MSP_EpmEnterpriseProjectType ON
          MSP_EpmProject_UserView.EnterpriseProjectTypeUID =
MSP_EpmEnterpriseProjectType.EnterpriseProjectTypeUID
```

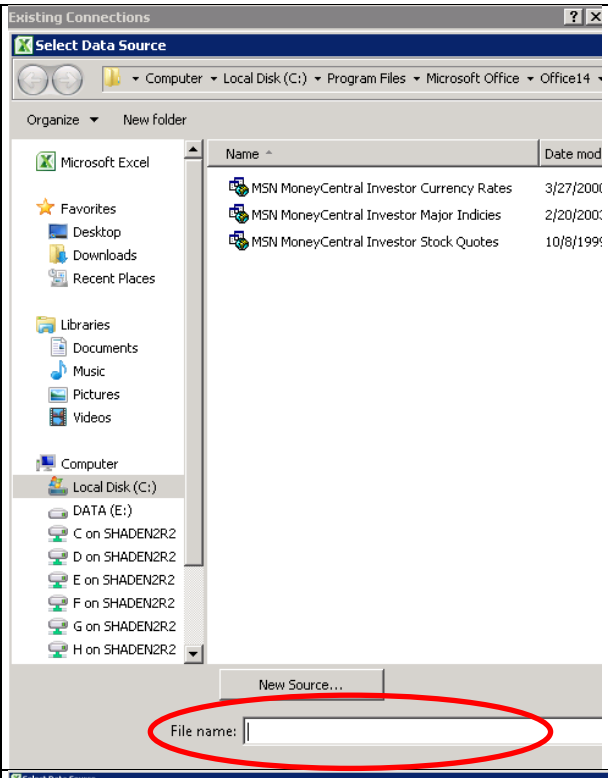
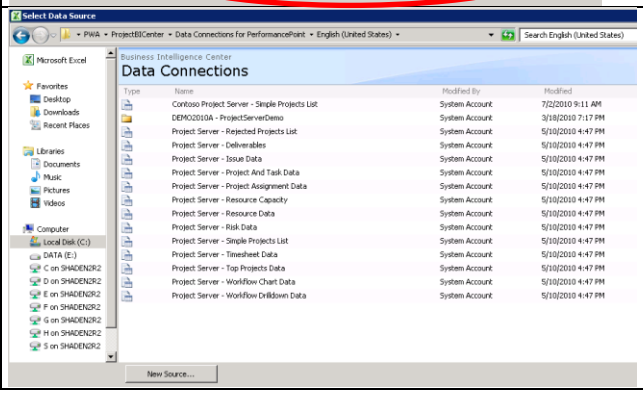


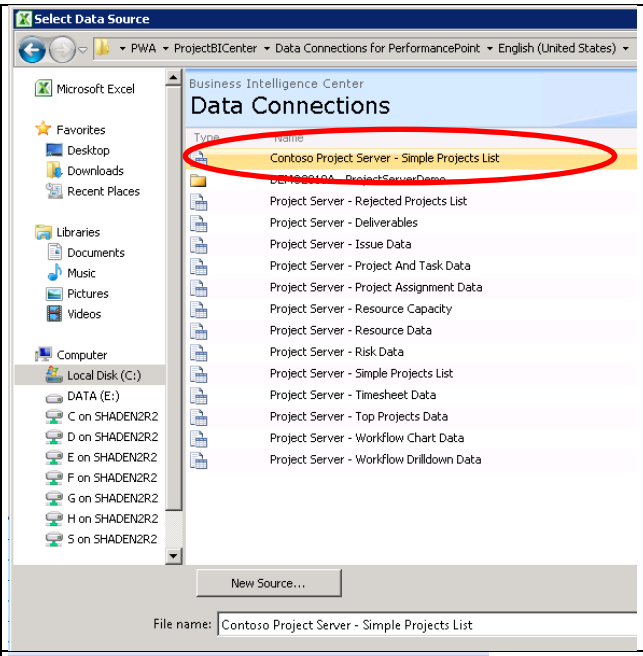
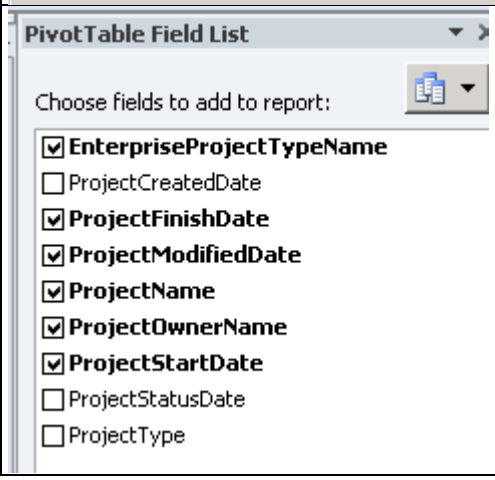
Upload the updated Connection file to the Data connections Library in the folder English (United States)	
Click the Upload Document button on the toolbar	
Select the file you have saved	

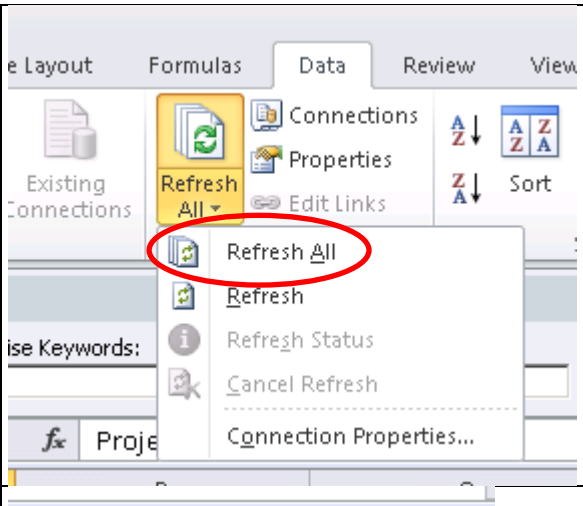
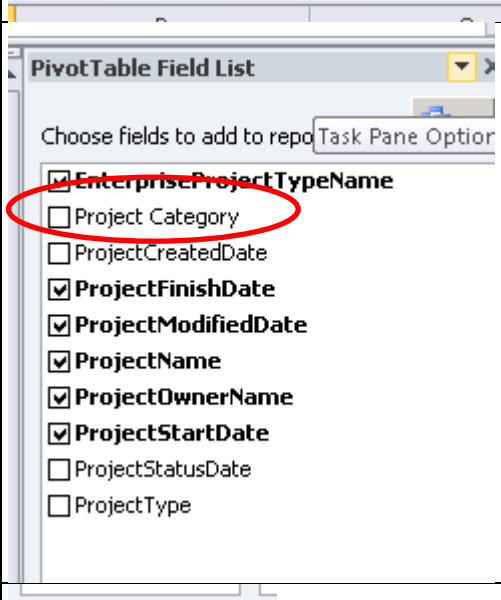
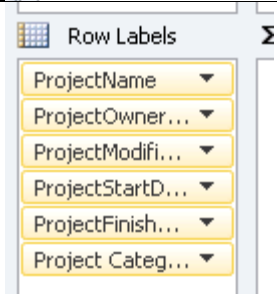
You need to update some fields on the property page. Content Type: Office Data Connection File Name: Contoso Project Server – simple Project Title: Contoso Project Server – Simple Projects List Description: This is an updated Data connection file for Contoso Keywords: Project, Enterprise Project Type, Project Server Reporting Click the Save button.	 
It is now present in the list of data connections in the library.	

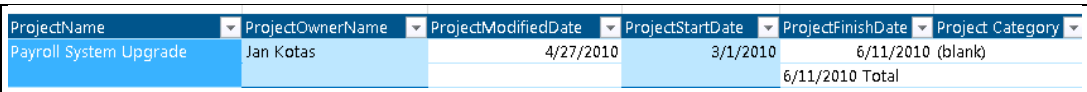
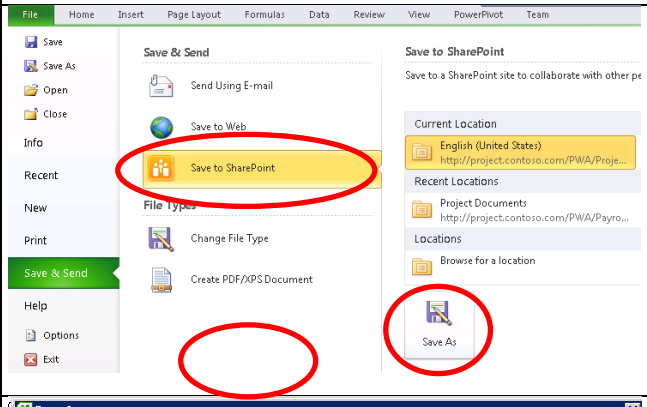
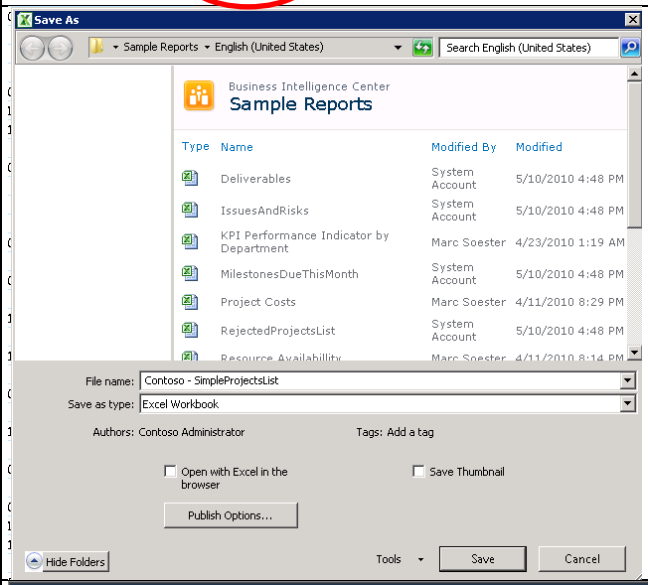
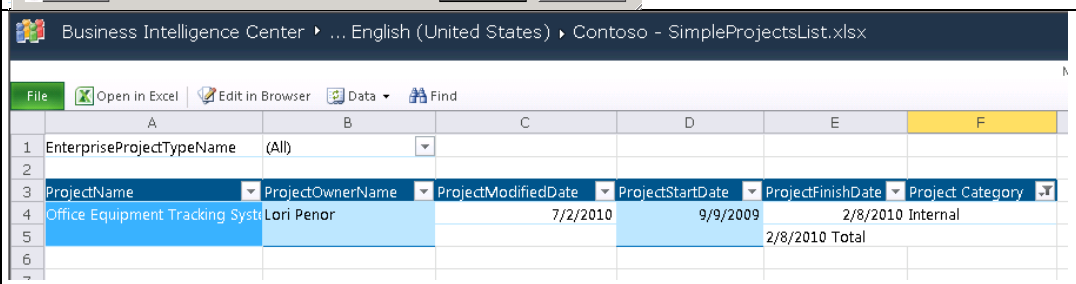
Publish it	
Select the URL of the data connection library: With the correct *<servername>* You can use the property page	http://<servername>/PWA/ProjectBICenter/Data Connections for PerformancePoint/English (United States)/ 
Open the Simple Project List report with Excel, On the Data tab, click the Connections button.	
Remove the current ODC file by using the Remove button.	
Validate the warning message by selecting	

the OK button.	
Add the new ODC connection to the current report. Click the Add button.	
A first dialog is presented to existing connections. Click the Browse for More button.	

Enter the URL: http://<servername>/PWA/ProjectBICenter/Data Connections for PerformancePoint/English (United States)/ in the File name box.	
Select the one we created	

	
We can check that the updated ODC is used by selecting the Refresh button to update the list of fields.	

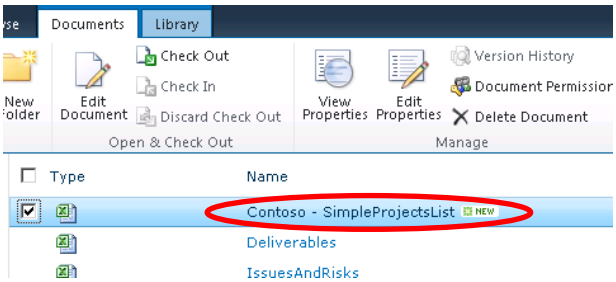
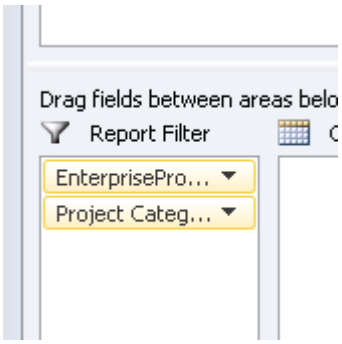
Click the Refresh All button in the Data tab, to refresh the list of fields.	
The field named Project Category is now present in the Field list	
Selecting the Project Category in the list of fields will add them to the Row Labels.	

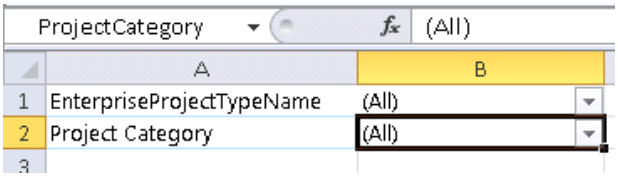
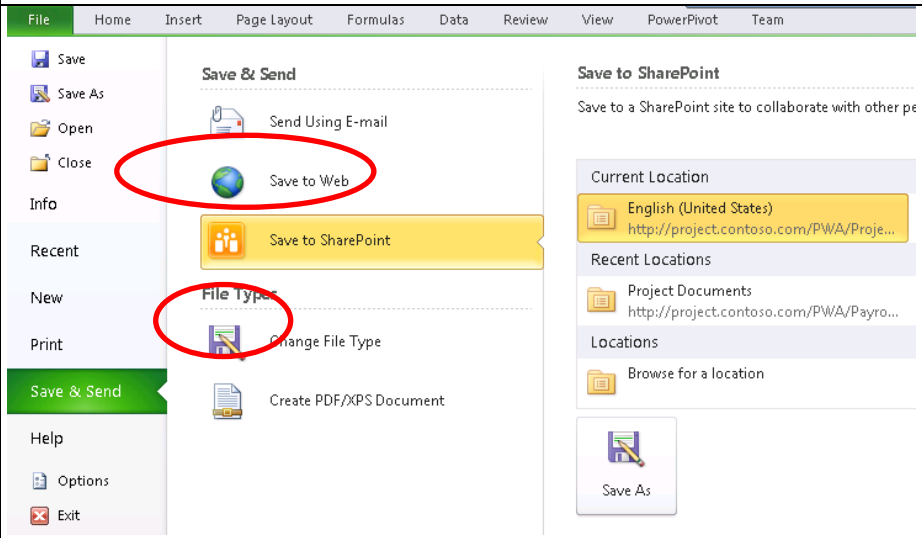
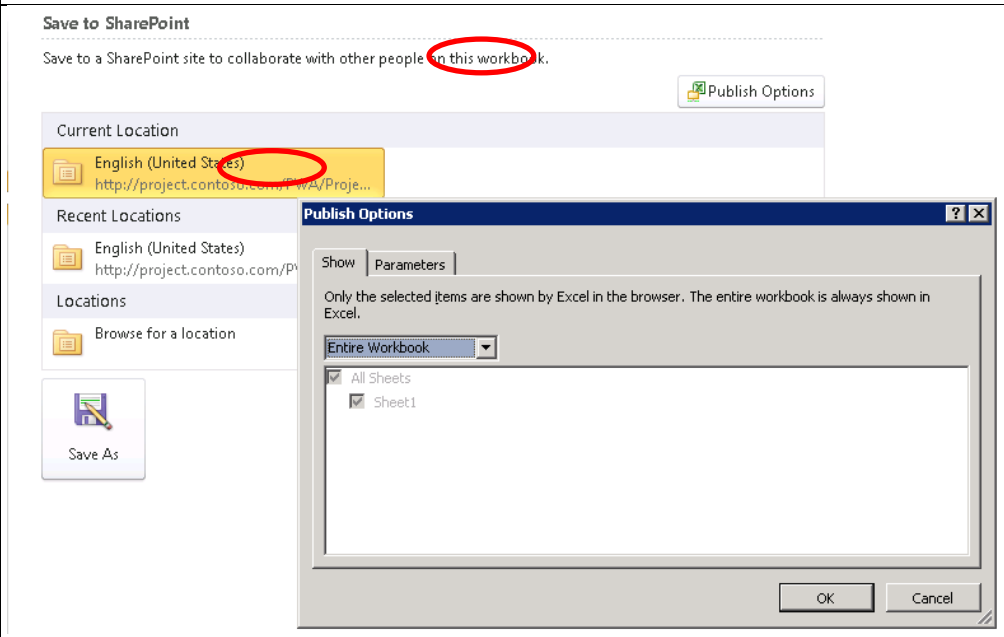
And in the report:	
Save the Report to a new name. Use the Menu Save & Send. Save to SharePoint Server and Select Save As.	
Enter the name Contoso Simple Project List. Clear the option named Open with Excel in the Browser.	
Here is the result of the report being filtered to show only the Internal project:	

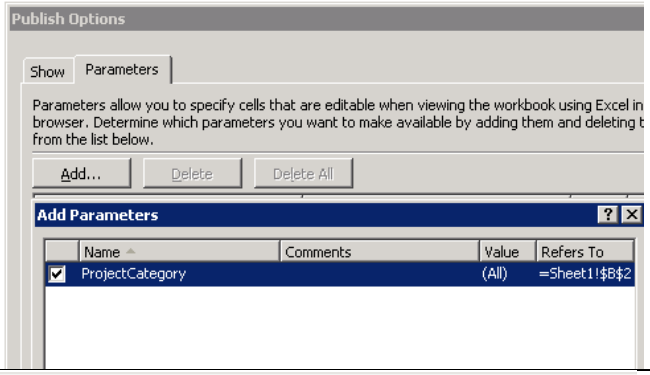
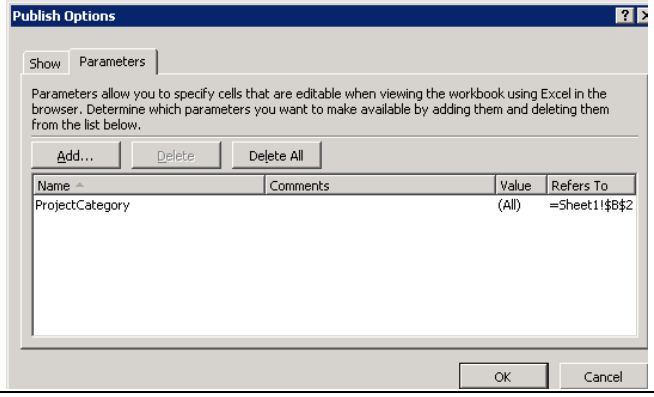
How to add a filter to the existing Report

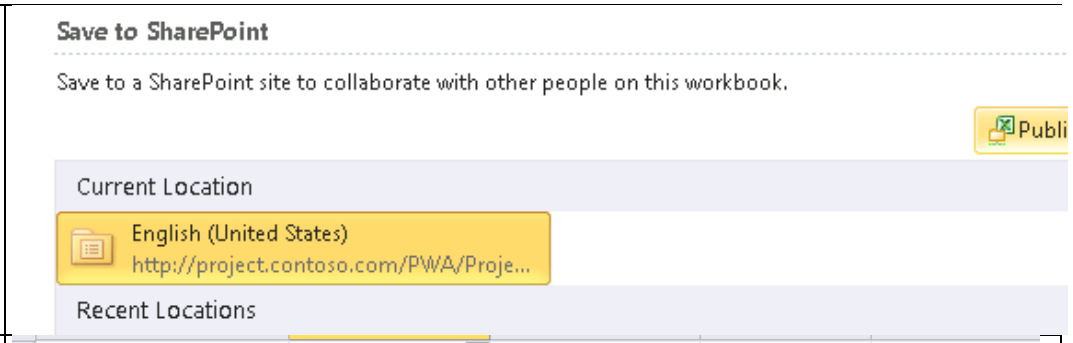
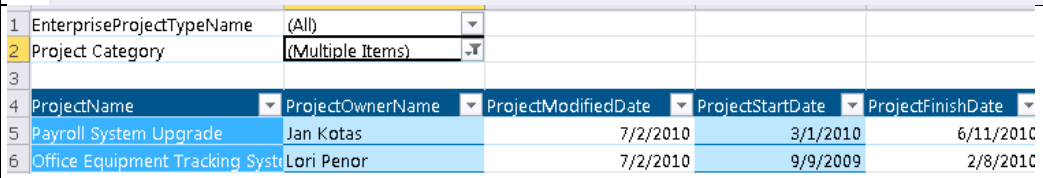
In this chapter we describe the precise steps involved in adding to our report a parameter to select the Project Category.

- We will update the created Report: Contoso Simple Project
- Publish this report

Actions	Screen
Navigate to the BI Center and select the sample Report: Contoso SimpleProjectList	
Select Edit with Excel.	
Add the field Project Category to the Report Filter.	

Name the Cell B2 ProjectCategory	
Save the report. Use the Menu Save & Send. Click Save to SharePoint.	
Select the Publish Options. In the new dialog box, select the Parameters tab.	

Click the Add button.	
Select the Project Category Parameter.	
Click OK.	

Select Current Location and save the report.	
Try the report.	
To fully use this filter, a dashboard should be built that would contain this report with this parameter, but that is beyond the scope of this document.	

Tips & Tricks

Do not modify default names

In the BI Center, we recommend that you NOT rename or delete the default content or its containing folders, to avoid maintenance problems when service packs are installed.

You can create your own folders with a specific naming convention (that includes your company name, for example Contoso Reports), so that you avoid name clash with future releases.

Microsoft reserves the right to update default content when patches or services packs are released.

Do not modify default ODC

The ODCs that are provided by Microsoft may be patched by subsequent Service Packs and Patches. Therefore, instead of modifying the predefined content, we recommend that you make a copy and then modify it.

5. Use PowerPivot for Excel 2010

Introduced with the SharePoint Server 2010 release, Power Pivot consists of both a SharePoint application ("Microsoft SQL Server PowerPivot for Microsoft SharePoint ") to generate list data feeds and an Excel add-in ("Microsoft SQL Server 2008 R2 PowerPivot for Microsoft Excel 2010 "). Power Pivot pulls data from SQL Server databases and SharePoint lists and easily aggregates it into a single table. (Power Pivot was deemed outside the scope of this document.)

Use this tool when multiple SharePoint lists must be combined with Project Server data to generate a single data set for reporting purposes. For example, use this tool to combine a list containing a project narrative from team members with project data – or if project-level metadata has been extended from Project Server into secure SharePoint lists.

Refer to the documentation in MSDN for additional details:
http://technet.microsoft.com/library/ff645392.aspx.

6. Migrating Reports between Environments

It is a good practice to design, build, and test you report on a Dev/Test environment before deploying it to a production environment.

Deployment of the Reports on a QA and Production environment

As we have seen the Excel Services are composed of two main objects:

- Workbook
- ODC files

The workbook is an .xslx file that can be easily managed.

Which files you have to deploy on the new environment depends on whether you are using a default Out of the Box ODC file or not.

You will also have to ensure that the same Custom Fields and Lookup tables are defined on both the Dev and AQ/production environment to ensure that the reports will run on both environments.

Here are the steps to deploy a report from Dev to a QA environment:

1. Check the configuration of Custom Fields and Lookup table, using the Playbooks tool for example or manage them manually,
2. Save the Report as an .xlsx workbook file from the Dev environment,
3. Export the ODC file if needed,
4. Upload the ODC file on the QA environment and update the SQL connection string
5. Upload the .xslx Workbook on the QA environment
6. Test your report

7. Troubleshooting Business Intelligence Features

This chapter gives a list of items to check when data is not displayed as expected in and Excel workbook when using Project Server reports.

Follow the steps given in this Blog article:

http://blogs.technet.com/b/projectadministration/archive/2009/12/15/troubleshooting-business-intelligence-features.aspx

Why isn't my workbook rendering on the web?

- One Time Setup Items
    - Has the Excel Services Service Application and Proxy Connection been created?
    - Has the Secure Store Service Application and Proxy Connection been created?
- Is the Excel Services Service Application configured?

    - Is the folder for the data connections in a trusted location within the Excel Services Service Application?
    - Is the folder for the reports in a trusted location within the Excel Services Service Application?
- Is the Secure Store Service (SSS) Application configured?
- For each Target Application Profile
    - Has the Secure Store Target Application Profile for the reports configured?
    - Is the user authorized or belongs to a group that is authorized to use the particular SSS Target Application profile?
    - Does the SSS Target Application have credentials set?
    - Does the SSS Target Application credentials have DB_DataReader rights to the Reporting database?
- For each workbook
    - Is the SSS ID value in the Office Data Connection filled in?
    - Does the SSS ID in the Excel workbook match the ID of the Application Profile in SSS?

Why hasn't my data appeared yet?

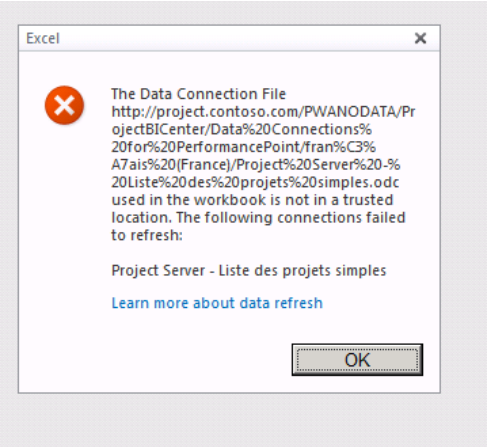
- If the data is sourced from the Reporting database, it could be that the Reporting Publish job has not yet completed. Because these jobs are queued, you must ensure that the job has completed before you will see the data.
- If the data is sourced from an OLAP database, you will not see the data until the OLAP database is refreshed.
- If these two items aren't the case, there may be an issue with the query itself.

Excel client is opened, instead of displaying in web page

Depending on the default configuration in the SharePoint Library that contains the Excel workbook when you open an Excel spreadsheet: Open in Browser/Call Excel Client

The following connection failed to refresh

If you forget to add the trusted configuration libraries to Excel Services, you will get this error message:



Cannot publish Excel to SharePoint

You are prompted to enter your credentials when you access an FQDN site from a computer that is running Windows Vista or Windows 7 and has no proxy configured; you may get an authentication prompt. This article explains how you update the configuration to support this configuration:

http://support.microsoft.com/kb/943280.

If you cannot publish directly to SharePoint Server, you can save the workbook to disk and upload the workbook in the SharePoint document library.

8. References

General references

Title	URL Reference
Project 2010 Web site	www.microsoft.com/project/2010
Project Server 2010 TechCenter (TechNet)	http://technet.microsoft.com/projectserver
Business Intelligence in Project Server 2010	http://go.microsoft.com/?linkid=9726143
Project 2010 Resource Center (MSDN®)	http://msdn.microsoft.com/Project
Project 2010 Video content	www.microsoft.com/showcase/en/US/channels/microsoftproject
Project 2010 webcasts and podcasts	www.microsoft.com/events/series/epm.aspx?tab=webcasts
Project 2010 Demo Image:	**Download:** http://go.microsoft.com/?linkid=9713956 **Hosted Virtual Lab:** http://go.microsoft.com/?linkid=9713654

Blogs	URL Reference
Official Blog of the Product Development group	http://blogs.msdn.com/project
Project Developer	http://blogs.msdn.com/project_programmability
Project IT Pro	http://blogs.technet.com/projectadministration
SharePoint BI Blog	http://blogs.technet.com/ppsdocteam
Report building with Excel Services	http://blogs.technet.com/ppsdocteam/archive/2009/04/29/report-building-with-excel-services.aspx

Additional questions? Project 2010 Forums!

http://social.msdn.microsoft.com/Forums/en-US/category/projectserver2010,projectprofessional2010

Existing White Papers

- Building a Report Based on Cube Functions for Excel Web Access
- Building an OLAP PivotTable Report with Excel Services
- Building a Relational PivotTable Report with Excel Services
- Building a Report with a Chart for Excel Web Access Based on Cube Functions

References given in this document

Document/Blog	URL
Troubleshooting Business Intelligence features in Project Server 2010	http://blogs.technet.com/b/projectadministration/archive/2009/12/15/troubleshooting-business-intelligence-features.aspx
OLAP Database and Analysis Services Configuration:	http://technet.microsoft.com/en-us/library/ee662108(office.14).aspx
Setup steps for Business Intelligence features	http://technet.microsoft.com/en-us/library/ee662106(office.14).aspx
How to create and	http://office.microsoft.com/en-us/excel/HA101672271033.aspx

modify Office Data Connections	
How to use Office Data Connections	http://blogs.msdn.com/excel/archive/2008/10/15/using-office-data-connection-files-odc-and-the-dataconnections-web-part-in-sharepoint-to-specify-external-data-connections-in-newly-created-excel-workbooks.aspx
Project Server 2010 SDK	http://www.microsoft.com/downloads/details.aspx?FamilyID=46007f25-b44e-4aa6-80ff-9c0e75835ad9&displaylang=en
Solution starters for Project Server 2010	http://code.msdn.microsoft.com/P2010SolutionStarter

9. List of Figures