



PRIMAVERA

**Cloud Services Reporting Administration Guide
Version 18**

April 2018

Contents

About This Guide.....	5
About Personally Identifiable Information	5
Reporting in P6 EPPM.....	5
P6 Publication Services	6
Assigning Permissions for P6 EPPM Reporting	7
Running a P6 Report.....	7
About P6 Publication Services and Reporting.....	8
Working with P6 Publication Services for Reporting.....	9
Defining Publication Periods	13
Publishing P6 Data for Reports	13
Enabling Automatic Publishing of P6 Project Data for Reports	14
Configuring Publication Service Settings for Projects	15
Enabling Automatic Publishing of P6 Global Data for Reports	15
Manually Publishing P6 Project Data for Reports	16
Manually Publishing P6 Global Data for Reports	17
Supported Parameters in P6	18
Enumeration Parameters.....	19
Dynamic Parameters.....	22
Primitive Parameters	28
Allowing for Multiple Values Returned via a List	29
Reporting in Primavera Unifier	29
Creating a BI Publisher Report in Primavera Unifier	29
Making a New Report Appear in Primavera Unifier.....	32
Assigning Permissions for Primavera Unifier Reports.....	32
Creating a Data View in Primavera Unifier.....	32
Creating a Custom Report in Primavera Unifier	34
Download and Install BI Publisher Desktop for Microsoft Office	34
Review Unifier Database Table Structure.....	35
Connect to Unifier Database	35
Writing Your First Statement	36
SQL JOIN	37
SQL WHERE	39
SQL ORDER BY Keyword	39
SQL Final Statement	40
Adding Line Items.....	42
SQL JOIN	43
SQL ORDER BY Keyword	43
SQL Final Statement	44
Creating a New Custom Report	44

Building Report Template	45
Creating a Report with Line Items.....	47
Formatting Data.....	48
Adding Summary Page to Report.....	49
Adding a Chart	50
Adding Headers and Footers	50
Adding BI Fields to RTF File Header or Footer	53
Importing Template into Unifier	54
Adding a Report to Navigator.....	54
Setting Permissions on Report	55
Running the Report	55
Advanced BI Publisher Functions.....	55
Search Parameters and Additional Parameters.....	56
Adding Permissions to Run BI Publisher Reports in Primavera Unifier	59
Running a Report in Primavera Unifier	59
Legal Notices	61

About This Guide

Scope

This guide contains all of the necessary information to use Oracle BI Publisher with your P6 EPPM and Primavera Unifier Cloud Service applications. BI Publisher is a reporting solution to efficiently author, manage, and deliver reports and documents.

For specific information about the tasks that are required to use BI Publisher with your Cloud Service application, see one of the following sections:

- ▶ **Reporting in P6 EPPM** (on page 5)
- ▶ **Reporting in Primavera Unifier** (on page 29)

Audience

This guide should be used by cloud administrators who have access to the following Cloud Service applications:

- ▶ Cloud Administration
- ▶ BI Publisher
- ▶ P6 EPPM or Primavera Unifier

About Personally Identifiable Information

Personally identifiable information (PII) is any piece of data which can be used on its own or with other information to identify, contact or locate an individual or identify an individual in context. This information is not limited to a person's name, address, and contact details, for example a person's IP address, phone IMEI number, gender, and location at a particular time could all be personally identifiable information. Organizations are responsible for ensuring the privacy of PII wherever it is stored, including in back-ups, locally stored downloads, and data stored in development environments.

Caution: Personally identifiable information (PII) may be at risk of exposure. Depending on local data protection laws organizations may be responsible for mitigating any risk of exposure.

Reporting in P6 EPPM

P6 relies on BI Publisher and the P6 EPPM Extended Schema to produce reports. To run reports in P6, you must complete these tasks:

- 1) For Publication Services, you will need to configure publication settings for General Data, Time Distributed Data, Blob Data, and Log retention. You can also configure publication services settings for Project Arbiter, Projects, Enterprise Data, Enterprise Summaries, Resource Management, and Security.
- 2) Configure and deploy the Publication Services required to populate the reporting views in the P6 EPPM Extended Schema. See ***Working with P6 Publication Services for Reporting*** (on page 9).
- 3) Configure BI Publisher to allow P6 to use parameter keys so users do not have to enter field values for reports.
- 4) When you begin working in P6, you can configure your application settings and global scheduled services. See the *P6 EPPM Application Administration Guide* for more information on scheduled services and application settings.
- 5) Provide users with the 'Edit EPS Costs/Financials' project privilege if they need to view project costs in a report generated from the P6 EPPM Extended Schema. See information about security profiles in the *P6 EPPM Application Administration Guide*.

P6 Publication Services

Data in the P6 Extended Schema is calculated and stored by Publication Services, which give administrators control over when data is calculated and stored in the P6 Extended Schema. Publication Services exist for project and global data. Global data includes enterprise data, as well as resource and role, portfolio, security and audit data. Project data includes all information about projects and baselines, including summaries and calculations.

The Publish Project service is dedicated to publishing project and baseline data to the P6 Extended Schema.

Note: The Publish Project service recalculates and publishes all project business objects containing calculated or denormalized data including, but not limited to: the WBS, activities, resource/role assignments, high-level planning assignments, relationships, risks, documents, expenses, steps, milestones, UDF values, notes, issues, budget change logs, timesheets, code assignments, funding source assignments.

These additional Publication Services publish global data to the P6 Extended Schema:

- ▶ Publish Audit Data
- ▶ Publish Enterprise Data
- ▶ Publish Enterprise Summaries
- ▶ Publish Resource Management
- ▶ Publish Security

The first time a data type is published to the P6 Extended Schema all data is recalculated and published to the extended schema tables. After a data type has been published for the first time, future publication services recalculate and publish only data that has changed since the last time that data was published.

When any of the publication services runs, each business object type processed by the service is individually timestamped, and the timestamp stored in the database. The services can accurately determine exactly which data has changed since the last time it was published using these timestamps because they exist at the individual business object type level.

Configure Publication Services in P6 as follows:

- ▶ Set up automatic publication and enable P6 to publish project and global data on the Services page in Application settings.
- ▶ Enable the types of global data to be scheduled in Global Scheduled Services.
- ▶ Disable or enable publication for a selected project on the Services page in Project Preferences.
- ▶ Add, verify the status of, modify, or delete project management services on the Project Scheduled Services page.

Tip

Run the global data services before turning on the Project Arbiter service.

Assigning Permissions for P6 EPPM Reporting

Complete the following steps to assign permissions for reporting:


- 1) Log in to Primavera Analytics.
- 2) Click **Catalog**.
- 3) Under **Shared Folders**, highlight the **P6Reports** folder.
- 4) In the **Tasks** pane, click **Permissions**.
- 5) Select the following checkboxes:
 - ▶ **Apply Permissions to sub-folders**
 - ▶ **Apply permissions to items within this folder**
- 6) Set permissions for the following roles:
 - ▶ **BI Consumer: Open**
 - ▶ **BI Content Author: Traverse**
 - ▶ **BI Service Administrator: Full Control**
- 7) Click **OK**.

Running a P6 Report

To run P6 reports, you must:

- ▶ Assign the P6 User and a BI Publisher (BIConsumer to run reports only; BI Author to create and run reports) application access to the user in Cloud Administration.
- ▶ Assign Enterprise Reports module access to the P6 user.
- ▶ Have a BI Publisher user that matches your P6 user name and has the P6 Reports role.
- ▶ Run Publication Services to ensure the report executes successfully.

To run reports:

- 1) Log in to P6.
- 2) Click **Reports**.
- 3) On the **Report** page, expand the report category and select a report.
- 4) Click **Run**.
- 5) On the **Options** tab of the **Run Report** dialog box, do the following:
 - a. In the **Schedule Name** field, enter a name for the report.
 - b. In the **Template** field, select a template to apply to the report.
 - c. In the **Delivery Type** list, select whether the report will be deliver as an email or file.
 - d. If you selected **Email** as the **Delivery Type**, click  **Email Options** and specify the email addresses for the report.
 - e. In the **Output Format** field, select the file format for the report.
 - f. Under **Notifications**, select the check boxes as needed.
- 6) On the **Parameters** tab of the **Run Report** dialog box, specify values for the parameters.

Note: When selecting Project parameters, if you choose to add an EPS to the Selected Projects list, only the projects directly under the EPS are added. If additional EPSs are under the parent EPS, these projects will not automatically be included in the report; however, you can select additional EPSs and add them to the Selected Projects list.

Caution: To avoid system performance issues, be specific when entering values for reports. Narrow down your choices to include only what is absolutely necessary.

- 7) Click **Run**.

About P6 Publication Services and Reporting

Before you can generate reports, data must be refreshed and calculations must be made in precisely arranged tables and fields. *Publication Services* address these reporting needs gathering and calculating data as close as possible to real-time. To make this happen, you can configure P6 to automatically publish data to reporting tables. Depending on the type of data published, P6 tables may still generate reports, or a combination of P6 and reporting tables may generate them. An administrator must configure settings for Publication Services before publishing reporting data.

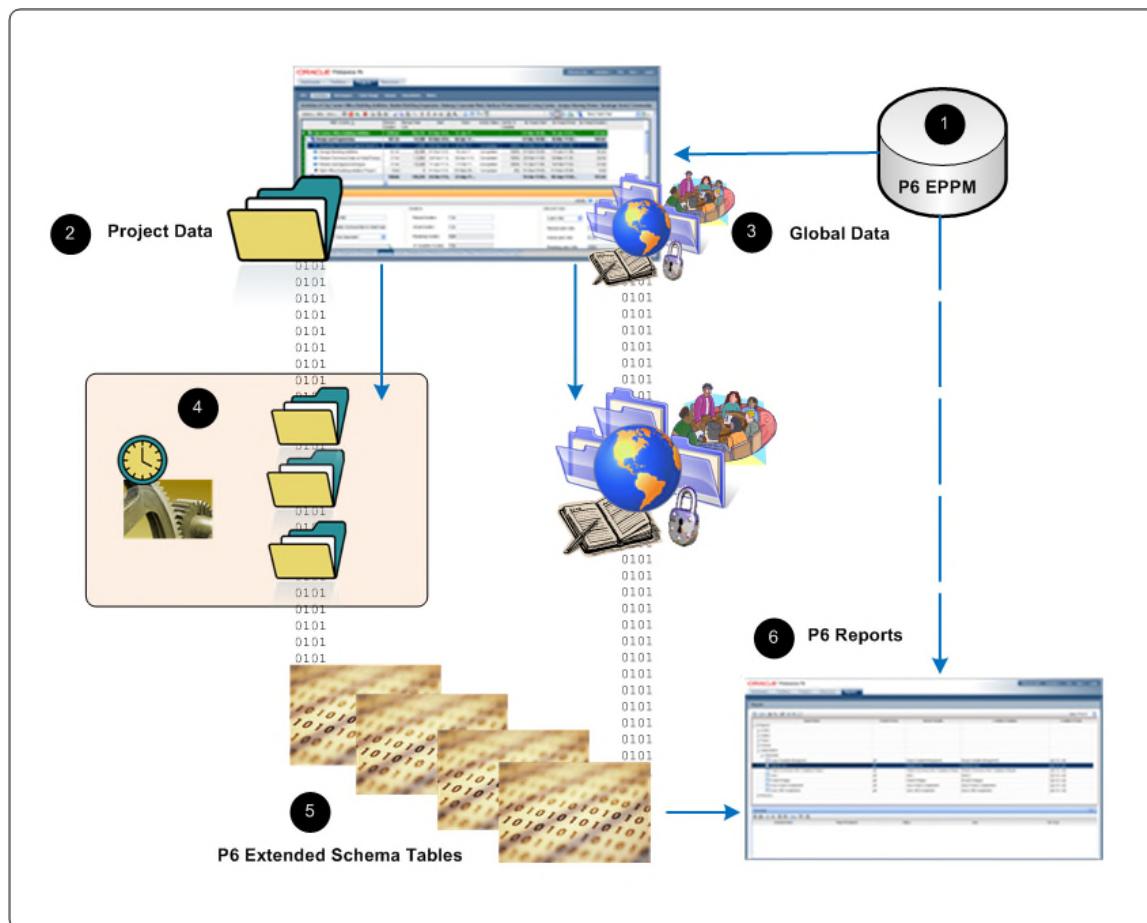
Your company's P6 EPPM data has two categories: *project* data and *global* data. Published project data includes all information about your projects, including summaries, calculations, and auditing data. Published global data includes enterprise data, resource and role data, portfolio data, security, and auditing data.

To summarize the reporting process:

- 1) Users create new data or make changes to existing data in P6.
- 2) P6 global and project data publishes, automatically or manually, to tables for reporting.
- 3) Reports generate directly from the collective P6 EPPM database with the P6 Extended Schema reporting tables.

Working with P6 Publication Services for Reporting

To achieve near real-time reporting, you can configure P6 to automatically publish to tables that store updated data, including calculations and summary data. These tables also create views for generating reports. The P6 data is split into two general categories: *project* data and *global* data. Project data includes all information about your projects, including summaries and calculations. Global data includes enterprise data, as well as resource and role, portfolio, and security data. The following image summarizes the key concepts in the publication and reporting process.



Item	Description
1	EPPM Database: The EPPM database captures all the data your organization generates every second of every day. However, the raw data in the EPPM database is not structured for immediate reporting.

2	Project Data: You can configure P6 to automatically publish a project based on publication thresholds. Based on the options you select, you can prioritize projects for publication in a queue. If necessary, you can manually add a project to the queue.
3	Global Data: You can configure services in P6 to schedule the publication of global data based on recurring intervals that you specify. If necessary, you can also immediately publish this data manually.
4	Project Queue: Projects publish in sequential order based on priority. Your administrator can control which projects are added to the publishing queue, when, and in what order. These options ensure the project data you need most is available for P6 reporting in near real-time.
5	<p>P6 Extended Schema Tables: The project and global services recalculate certain logical fields in the P6 EPPM database and store them as physical fields in the P6 Extended Schema tables so they are available for reporting and other purposes. Some fields in P6, such as durations, are calculated in real time as related field data is changed and are not stored in the database. Publication services recalculate these fields and store their values in the extended schema tables. Other fields, such as note fields, are not stored in the database in a format suitable for reporting. For these fields, publication services will convert them to a format more suitable for reporting.</p> <p>Note: For information about which tables are updated by the Publication Services for Reporting, see the following knowledge management document:</p> <p>What PX tables are updated when running Global Scheduled Services or Publish Project [ID 1491245.1]</p> <p>https://support.oracle.com/epmos/faces/ui/km/DocumentDisplay.jspx?id=1491245.1</p>

6

P6 Reports: You can run reports directly against P6 Extended Schema table views.

ORACLE PRIMAVERA P6 EPPM Welcome, Ruth Jones Administer Print Help Logout

Dashboards Portfolios Projects Resources

Application Settings

Audit
Data Limits
Earned Value
Gateway
General
ID Lengths
Reports
Services
Timesheets
Time Periods

Services

Publication

Publication Period

Specify the time period and interval for publishing time-distributed data.

Start date *

Finish date is current date plus

Between 0 and 99.

Time distributed interval

Day

Project Publication

☒ Enable Publish Projects

Publish projects every

Between 1 and 99.

Publish a changed project when the...

Number of changes exceeds

Between 1 and 999.

Time since last publication exceeds *

8h

☐ Publish idle projects

Maximum number to publish

Between 0 and 999.

☒ Enable Baseline Publication

Summarization

Specify the interval to summarize and store resource spreads.

☒ Summarize by Calendar

WBS Level

Week

Resource/Role Assignment

Week

☐ Summarize by Financial Periods

Server Location for Scheduled Services

Specify the default server location to import and export files

Default Location

C:\PRIMAVERA\P6EPPM152\p6ExportedFiles

Project Preferences of Harbour Pointe Assisted Living Center

Analytics
Calculations
Contract Management
Defaults
General
Integration
Unifier
Services
Team Member

Services

Summarization

☒ Enable Summarization

Summarize project based on

☐ High level resource planning

☒ Detailed activity resources assignments

Summarize to WBS Level

All

Last Summarized Date 04-Aug-15 04:34 PM

Publication

☒ Enable Publication

Publication Priority 50

Last Published On 05-Aug-15 06:26 PM

Primavera P6 - Global Scheduled Services - Mozilla Firefox

Global Scheduled Services

Name	Enabled	Status	Last Run	Next Run
Publish Security	<input checked="" type="checkbox"/>	Completed	04-Aug-15 07:30 PM BST	05-Aug-15 07:30 PM BST
Publish Enterprise Data	<input checked="" type="checkbox"/>	Completed	04-Aug-15 07:31 PM BST	05-Aug-15 07:30 PM BST
Publish Enterprise Summaries	<input checked="" type="checkbox"/>	Completed	04-Aug-15 07:30 PM BST	05-Aug-15 07:30 PM BST
Publish Resource Management	<input checked="" type="checkbox"/>	Completed	04-Aug-15 07:30 PM BST	05-Aug-15 07:30 PM BST
Publish Audit Data	<input checked="" type="checkbox"/>	Completed	04-Aug-15 07:30 PM BST	05-Aug-15 07:30 PM BST

Publish Enterprise Summaries Settings

Run Service Daily

Start Time 09:30 PM BST

Item	Description
1	Application Settings for Publication Services: Located under the Administer menu, begin by clicking Application Settings , and then choose Services . On the Services page, specify the start date, period of time to add to the current publication date to determine a finish date, and time-distributed interval for publication. In the Project Publication section, you can enable the <i>Publish Projects</i> services to refresh project data according to a time interval you set. You can further determine a change threshold, time threshold, and even decide if idle projects should be added to the queue, and if so, set a maximum limit to the concurrent number of idle projects being published.
2	Project Preferences for Publication Services: Then, navigate to the EPS page in the Projects section. Click a project and select Set Project Preferences from the Actions menu. On the Services page, select the Enable Publication option to indicate you want to include this project in the <i>Publish Projects</i> services. All new projects are enabled for publication by default. Clear the option if you do not want to publish the project's data. On this page, you can also set the project's priority relative to other projects entering the publication queue.
3	Global Scheduled Publication Services: Return to the Administer menu and navigate to the Global Scheduled Services dialog box to enable the types of global data services you want to schedule. Below each service you enable, specify the recurring interval and start time in the detail window.

Defining Publication Periods

Before publishing global or project data for reporting, define the publication period. The publication period you define should be large enough to capture all the project data you need to include in reports.

Caution: You can perform these steps again at any time to modify your settings. However, if an administrator modifies the **Start date**, **Time distributed interval**, or **Finish date is current date plus** fields, all data will be fully recalculated. Oracle recommends that you only reconfigure publication options on weekends or during off-peak hours to prevent the disruption of other P6 features.

To define the publication period:

- 1) From the **Administer** menu, select **Application Settings**.
- 2) On the **Application Settings** pane, click **Services**.
- 3) On the **Services** page, in the **Publication Periods** section:
 - a. In the **Start date** field, browse to the exact month, day, year, and time to mark the initial start of the data publication period. Select any date in the past that represents a reasonable amount of historical spread data for your organization. For example, in order for users to be able to produce time-distributed reports for any date range, enter the earliest project start date at your organization.
 - b. In the **Finish date is current date plus** field, select a numeric quantity and unit of time to construct a dynamic period of time. Whenever a publication service runs, this period of time is added to the current date to determine the finish date for the publication of data. For example, if the value is 5 years, time-distributed data will always be published covering the period of time that begins with the value in the **Start Date** field and extends five years into the future each time a service runs.
 - c. In the **Time distributed interval** field, select the unit of time in which time distributed data will be recalculated and published. **Time distributed interval** is set to Week by default. Set this to Day if you need to see daily data.
 - d. Click **Save**.

Publishing P6 Data for Reports

Reports based on P6 data require timely access to the most current data in order to be accurate. For example, report recipients expect to see updated enterprise data and project data including any calculations. In order to provide this data, P6 generates and stores data in reporting tables (the *P6 Extended Schema*). You can configure P6 to automatically publish essential data to the tables or manually generate it so that the most current data is available for reporting.

The data is split into two general categories: *project* data and *global* data. Project data includes all information about your projects, including baselines, summaries, calculations, and audit data and global data includes enterprise data, resource and role data, security and audit data.

Enabling Automatic Publishing of P6 Project Data for Reports

Perform the following procedure to enable projects for publication, and to set options for automatic project publication. Then, as you work, P6 automatically detects the changes to your projects that trigger the publication of their data.

To enable Publication Services for project data:

- 1) From the **Administer** menu, select **Application Settings**.
- 2) On the **Application Settings** pane, select **Services**.
- 3) On the **Services** page, in the **Project Publication** section:
 - a. Select **Enable Publish Projects** to enable automatic project publication based on defined thresholds. Selecting this option also allows users to manually publish projects.
 - b. In the **Publish projects every** field, select an interval by which projects are polled to be published.
 - c. Enter a number in the **Number of changes exceeds** field. This threshold setting determines the number of edits users can make to the project data before P6 publishes its data. Assuming a constant rate of change among projects, a lower value will result in more frequent publication of project data. If you enter a value of 0, projects with tracked changes will be automatically published.
 - d. Enter a time period for the **Time since last publication exceeds** field. This threshold setting determines how often the publication of project data should occur. For example, if you enter 12 hours, the project data will be published every 12 hours unless the threshold for the number of changes has already been reached.
 - e. Select **Publish idle projects** to add projects to the service queue that are enabled for publication but have not been changed during the time threshold. This setting is only valid for the initial run of the service.
 - f. In the **Maximum number to publish** field, enter the maximum number of pending idle *Publish Project* services that can be present at once in the service queue. This prevents performance problems during peak demand when enabling the publication of a large number of projects. For example, if the service runs and queues 40 projects that have exceeded specified thresholds and must be published, or that have been manually published, and you have set the maximum to 100, P6 will schedule up to 60 idle projects for publication.
 - g. Select **Enable Baseline Publication** if you want to be able to publish baseline data.
 - h. Click **Save**.

Tips

- ▶ If your organization is upgrading to P6, select **Publish idle projects** to add your migrated projects to the service queue after your database is upgraded. This will publish all your projects in the queue and refresh the available data for reporting. After an upgrade, this setting is no longer applicable, and projects will be submitted to the service queue based on threshold values specified on the **Application Settings** page.
- ▶ Clear the **Publish idle projects** option if your organization does not report against completed projects.

- ▶ The **Maximum number to publish** field is only applicable immediately following an upgrade, when all projects are considered idle. When all projects have been published, the service queue will no longer be constrained based on this setting.
- ▶ A project will be automatically submitted to the service queue if you change the project baseline, calendar, or data date. Projects are also automatically submitted to the queue if you select or clear **Calculate Activity % Complete from activity steps** option, or modify the default price/unit value for assignments without activities, from the **Set Project Preferences** dialog box available from the **Actions** menu on the **EPS** page.
- ▶ For threshold settings, projects are added to the service queue based on tracked changes to data. Only changes to activities, resource/role assignments, relationships, and the WBS are tracked.

Configuring Publication Service Settings for Projects

After enabling automatic publishing, perform the following procedure to configure settings for each individual project in *Publication Services*. Then, as you work, P6 automatically detects the changes to your project that trigger the publication of its data in the service queue.

To configure Publication Service settings for a project:

- 1) Click **Projects**.
- 2) On the Projects navigation bar, click **EPS**.
- 3) On the **EPS** page, complete the following:
 - a. Select a project.
 - b. From the **Actions** menu, select **Set Project Preferences...**
- 4) On the **Project Preferences** pane, select **Services**.
- 5) In the **Publication** section, on the **Services** page, complete the following:
 - a. Select the **Enable Publication** option.

Note: By default, all projects are enabled for publication. The **Enable Publication** setting only needs to be modified if you wish to disable publication for a project or re-enable publication of a previously disabled project.

- b. Adjust the relative **Publication Priority** up or down between 1 and 100 with 50 being the default priority value, 1 being the highest priority, and 100 being the lowest.
 - c. Click **Save and Close**.

Tips

- ▶ If you wish to modify publication settings for many projects at one time, you can display the **Enable Publication**, **Last Published On**, and **Publication Priority** fields as columns in your EPS view.
- ▶ If you want to publish Baseline data, the project must be published.

Enabling Automatic Publishing of P6 Global Data for Reports

Perform the steps below to configure P6 to automatically publish any of the following types of global data to reporting tables.

- ▶ Enterprise Data
- ▶ Enterprise Summary Data including portfolio data
- ▶ Resource Management Data
- ▶ Security Data
- ▶ Audit Data

To automatically publish P6 global data:

- 1) From the **Administer** menu, select **Global Scheduled Services**.
- 2) In the **Global Scheduled Services** dialog box, complete the following
 - a. Select a service, then configure its settings in the **Service Settings** detail window. For example, you might specify that the service runs daily on Wednesdays with a start time of 10:15 PM.

Note: Oracle recommends running the **Publish Security** service first if the **Run After Previous** option is selected in the **Run Service** list for other publication services. Running the **Publish Security** service first will ensure that security data updates in the extended schema as soon as possible and ensures that the security restrictions are in place before you run the report.

- b. Select the **Enabled** option for any of the four global services listed.
 - c. If you choose to run one or more services with the relative frequency value of *After previous service*, click **Move Up** (Ctrl+Alt+Up) or **Move Down** (Ctrl+Alt+Down) to arrange the services in your preferred sequence.
 - d. Click **Save**.
 - e. Click **Close**.

Manually Publishing P6 Project Data for Reports

Perform the steps below to publish the data from one or more projects to the reporting tables so it is ready for reporting. Any of the projects' baselines which is enabled for publishing will also be published when you perform these steps. P6 will automatically publish project data; however, you may want to publish the data manually in special cases such as when generating an important report at a specific time.

To manually publish P6 project data:

- 1) Click **Projects**.
- 2) On the **Projects** navigation bar, click **EPS**.
- 3) On the **EPS** page, select one or more projects, or select an EPS node to publish all of its subordinate projects.
- 4) On the **Actions** menu, click **Publish Projects**.

Tips

- ▶ In order to manually publish project data from the EPS page, *Publication Services* must be enabled and configured.

- ▶ You can publish all open projects by clicking **Publish Projects** on the **Run** submenu on the **Actions** menu of the **Activities** page. You can also publish projects by selecting one or more projects on the **EPS** page, and then using the right-click menu.
- ▶ Projects are not immediately published. Instead, they will be added to the queue of projects being processed for publication the next time the Publish Projects service runs. View settings for the Publish Projects service, including how often the service runs, on the **Services** page, which is located on the **Application Settings** pane. Depending on your security privileges, the **Application Settings** pane may not be accessible. You can check the status of the Publish Projects service after it has been added to the service queue by selecting **View Service Status** from the **Actions** menu on the **Activities** page.
- ▶ If some of the selected projects are not enabled for publication, only those projects enabled for publication will be submitted to the service queue. If none of the selected projects are enabled for publication, you will not be able to click **Publish Projects**.
- ▶ A project will be automatically submitted to the service queue if you change the project baseline, calendar, or data date. Projects are also automatically submitted to the queue if you select or clear the **Calculate Activity % Complete from activity steps** option, or modify the default price/unit value for assignments without activities from the **Set Project Preferences** dialog box available from the **Actions** menu the **EPS** page.

Manually Publishing P6 Global Data for Reports

Perform the steps below to manually publish any of the following types of global data to the reporting tables.

- ▶ Enterprise Data
- ▶ Enterprise Summary Data including portfolio data
- ▶ Resource Management Data
- ▶ Security Data
- ▶ Audit Data

P6 will automatically publish global data; however, you may want to publish the data manually in special cases such as when generating an important report at a specific time.

To manually publish P6 global data:

- 1) From the **Administer** menu and select **Global Scheduled Services**.
- 2) In the **Global Scheduled Services** dialog box:
 - a. Select any of the four global services listed.
 - b. Click **Run Service** on the **Global Scheduled Services** toolbar.
 - c. In the resulting message box, click **OK**.
 - d. Click **Save**.
 - e. Click **Close**.

Tips

- ▶ If the service listed under the manually selected service is configured to run **After previous service**, it will run automatically when the selected service finishes.
- ▶ You must have the global security privilege, 'Administer Global Scheduled Services,' to run a service from the **Global Scheduled Services** dialog box.

Supported Parameters in P6

You can use any parameter for BI Publisher, but users will have to ensure they enter the value correctly, or the SQL statements in the data template will fail. Using supported parameters will allow you to provide a user interface to enter the values for parameters. Supported parameters for P6 fall into one of three categories:

- ▶ **Enumeration** parameters allow users to pick parameters from static lists.
- ▶ **Dynamic** parameters will generate the list at run time. For example, when focusing on Project ID, users will see a Project list that pulls the current projects from the P6 EPPM database.
- ▶ **Primitive** parameters support basic selections. For example, if a parameter is a Boolean, an option will appear for users to select or clear a text box instead of typing true or false.

The following are the supported parameters for P6, grouped by category:

Enumeration Parameters

- ▶ Activity Priority
- ▶ Assignment Proficiency
- ▶ Activity Status
- ▶ Activity Type
- ▶ Constraint Type
- ▶ Duration Type
- ▶ Percent Complete Type
- ▶ Project Status
- ▶ Rate Source
- ▶ Rate Type
- ▶ Resource Type
- ▶ Risk Status
- ▶ Risk Type

Dynamic Parameters

- ▶ Activity Code Value
- ▶ User Defined Activity Code (type and value)
- ▶ Cost Account
- ▶ EPS
- ▶ Expense Category
- ▶ Funding Source
- ▶ Portfolio
- ▶ Project
- ▶ Project Code Value
- ▶ User Defined Project Code (type and value)
- ▶ User Defined Resource Code
- ▶ Resource Code Value
- ▶ Resource Team

- Resource ID
- Responsible Manager
- Risk Category
- Role
- Role Team
- Timesheet Period
- User

Primitive Parameters

- Date
- Boolean
- Integer
- Float

Custom Parameters

- List of Values

Enumeration Parameters

Enumeration parameters map to fields that have a set list of possible values. When running a report with an enumeration parameter, P6 will offer a list to select one of the values. Users will see the localized text for the enumeration value's description; however, the return value will be the English description of the enumeration, which is the value the PX Views database stores.

This section contains the following information for each supported enumeration parameter:

- **Identifier:** The value you must enter in the identifier field in BI Publisher when creating the parameter for the report.
- **Details:** A description and technical details of the parameter.
- **Values:** The values that will be available in P6.
- **Maps to field:** The database field the return value maps to in the PX Views database. It could match multiple fields in the database, so only the primary table is listed.
- **Use case:** An example of how you might use the parameter in a report.

The following enumeration parameters are supported:

Parameter: Activity Priority

- **Identifier:** p_activity_priority
- **Details:** Allows users to select the leveling priority of an activity.
- **Values:** Top, High, Normal, Low, Lowest
- **Maps to field:** ACTIVITY.LEVELINGPRIORITY
- **Use case:** Filter activity data by leveling priority.

Parameter: Assignment Proficiency

- **Identifier:** p_assignment_proficiency
- **Details:** Allows users to select a value for assignment proficiency.

- **Values:** Master, Expert, Skilled, Proficient, Inexperienced
- **Maps to field:** RESOURCEASSIGNMENT.PROFICIENCY
- **Use case:** Filter resource assignment data by the proficiency of the assignment.

Parameter: Activity Status

- **Identifier:** p_activity_status
- **Details:** Allows users to select activity status.
- **Values:** Not Started, In Progress, Completed
- **Maps to field:** ACTIVITY.STATUS
- **Use case:** Filter activity reports based on a certain status.

Parameter: Activity Type

- **Identifier:** p_activity_type
- **Details:** Allows users to select the activity type.
- **Values:** Task Dependent, Resource Dependent, Level of Effort, Start Milestone, Finish Milestone, WBS Summary
- **Maps to field:** ACTIVITY.TYPE
- **Use case:** Filter activity reports based on the type of the activity.

Parameter: Constraint Type

- **Identifier:** p_constraint_type
- **Details:** Allows users to select an activity constraint type.
- **Values:** Start On, Start On or Before, Start On or After, Finish On, Finish On or Before, Finish On or After, As Late As Possible, Mandatory Start, Mandatory Finish
- **Maps to field:** ACTIVITY.PRIMARYCONSTRAINTTYPE and ACTIVITY.SECONDARYCONSTRAINTTYPE
- **Use case:** Filter activities in a report by the activity primary or secondary constraint type.

Parameter: Duration Type

- **Identifier:** p_duration_type
- **Details:** Allows users to select the duration types of an activity.
- **Values:** Fixed Units/Time, Fixed Duration & Units/Time, Fixed Units, Fixed Duration & Units
- **Maps to field:** ACTIVITY.DURATIONTYPE
- **Use case:** Filter activities in a report based on their duration type.

Parameter: Percent Complete Type

- **Identifier:** p_percent_complete_type
- **Details:** Allows users to select the percent complete type of an activity.
- **Values:** Physical, Duration, Units
- **Maps to field:** ACTIVITY.PERCENTCOMPLETETYPE
- **Use case:** Filter activities in a report based on the percent complete type of the activity.

Parameter: Project Status

- **Identifier:** p_project_status
- **Details:** Allows users to select the status of a project.
- **Values:** Planned, Active, Inactive, What If, Requested, Template
- **Maps to field:** PROJECT.STATUS
- **Use case:** Filter the projects in a report based on the desired type. For example, you might want to use this for a report that needs to display information only on planned projects.

Parameter: Rate Source

- **Identifier:** p_rate_source
- **Details:** Allows users to select the rate source of an assignment.
- **Values:** Resource, Role, Override
- **Maps to field:** RESOURCEASSIGNMENT.RATESOURCE
- **Use case:** Filter resource assignments included in a report based on the rate source of the assignment.

Parameter: Rate Type

- **Identifier:** p_rate_type
- **Details:** Allows users to select the rate type of an assignment.
- **Values:** Price/Unit, Price/Unit 2, Price/Unit 3, Price/Unit 4, Price/Unit 5
- **Maps to field:** RESOURCEASSIGNMENT.RATETYPE
- **Use case:** Filter resource assignments included in a report based on the rate type of the assignment.

Parameter: Resource Type

- **Identifier:** p_resource_type
- **Details:** Allows users to select the resource type of an assignment.
- **Values:** Labor, Nonlabor, Material
- **Maps to field:** RESOURCEASSIGNMENT.RESOURCE TYPE
- **Use case:** Filter resource assignments included in a report based on the resource type of the assignment.

Parameter: Risk Status

- **Identifier:** p_risk_status
- **Details:** Allows users to select the status of a risk.
- **Values:** Proposed, Open, Active, Rejected (Closed), Managed (Closed), Impacted (Closed)
- **Maps to field:** RISK.RISKSTATUS
- **Use case:** Filter risks in a report based on the status of the risk.

Parameter: Risk Type

- **Identifier:** p_risk_type

- **Details:** Allows users to select the type of a risk.
- **Values:** Threat, Opportunity
- **Maps to field:** RISK.RISKTYPE
- **Use case:** Filter risks in a report based on the type of risk.

Dynamic Parameters

Dynamic parameters map to fields that have a varying list of possible values. When running a report with a dynamic parameter, P6 will offer a list to select one of the available values.

This section contains the following information for each supported dynamic parameter:

- **Identifier:** The value you must enter in the identifier field in BI Publisher when creating the parameter for the report. In some cases, identifiers can pass in context by appending short names to the end of the identifier. The character in quotations is the separator that the code splits and <name> represents the context you are trying to pass in.
- **Details:** A description and technical details of the parameter.
- **P6:** What the editor for the parameter will be in the Reports section of P6.
- **Return Value:** The values that will be available in P6.
- **Maps to field:** The database field the return value maps to in the PX Views database. It could match multiple fields in the database, so only the primary table is listed.
- **Use case:** An example of how you might use the parameter in a report.

The following dynamic parameters are supported:

Parameter: Activity Code Value

- **Identifier:** p_a_code_val__<short name>
- **Details:** Enables users to select an activity code value via a list. Context passes into the parameter by appending a colon":" followed by the short name of the activity code type you want to set.
- **P6:** Provides a list that displays the Activity Code Values for the Activity Code type passed in the context.
- **Return value:** Short name of the Activity Code (unique per code type).
- **Maps to field:** ACTIVITYCODE.CODEVALUE
- **Use case:** Create a report that displays some basic information about activities. Users at five locations need to run the report, but they only want to see the data for activities with codes matching their location. Instead of creating five reports hard coding the location (for example, location=L1) on each report, you can create one report and add this parameter to it (for example, p_a_code_val__Location).

In the data template for the report, filter the activities based on this parameter. Hard code the left side of the activity filter to match the activity code you selected, which in this case is location.

Example query: CODETYPENAME='Location' &
CODEVALUE=:p_a_code_val__Location

If you did not have a parameter for this, you would need different templates for each location.

Parameter: User Defined Activity Code

- ▶ **Identifier:** p_activity_code_value "." <number> p_activity_code_type "." <number>
- ▶ **Details:** Enables users to select a user defined Activity Code. The user defined Activity Code is two parameters on the report in BI Publisher, but will display only as one row in the report settings parameter table.
- ▶ **P6:** Provides a list that displays all global Activity Code types. When users expand a type, the list shows the values for that type. By selecting a value, both the type and value will return to the report.
- ▶ **Return value:** Short name for the Activity Code Value, primary key for the Activity Code type.
- ▶ **Maps to field:**
 - p_activity_code_value maps to ACTIVITYCODE.CODEVALUE
 - p_activity_code_type maps to ACTIVITYCODE.CODETYPEOBJECTID
- ▶ **Use case:** Create a report that can have a variable Activity Code. The report pulls activities and displays some basic statistics of the activities. The data template for the report must accommodate setting both sides of the query. While a typical parameter just sets the IN clause for a user defined field, this parameter must set both sides. The "Activity Code Value" parameter **Use case** example shows where it hard codes the CODETYPENAME to be **Location**. This parameter enables multiple user defined activity codes to be used on the same report. For each parameter you use, you must add p_activity_code_value.1 and p_activity_code_type.1. There must be a pair of numbers to ensure that the editor works properly.

Parameter: Cost Account

- ▶ **Identifier:** p_cost_account
- ▶ **Details:** Enables users to select Cost Accounts. The P6 user must have access to view Cost Accounts for the list to populate.
- ▶ **P6:** Provides a list that displays all Cost Accounts in a hierarchical tree.
- ▶ **Return value:** Short name of the cost account (unique).
- ▶ **Maps to field:** COSTACCOUNT_FULL.NAME
- ▶ **Use case:** Filter items using certain Cost Accounts or generate information on the Cost Accounts.

Parameter: EPS

- ▶ **Identifier:** p_eps_id
- ▶ **Details:** Enables a user to select an EPS.
- ▶ **P6:** Provides a list that displays all of the EPS nodes where the user has access.
- ▶ **Return value:** The short name of the EPS.
- ▶ **Maps to field:**
 - EPS_FULL.NAME
 - EPS_U.NAME
- ▶ **Use case:** Use a parameter for EPS to filter a query to load all projects under an EPS for a report.

Parameter: Expense Category

- **Identifier:** p_expense_category
- **Details:** Enables a user to select Expense Category where the user has access.
- **P6:** Will provide a list that displays all of the Expense Categories.
- **Return value:** The short name of the Expense Category (unique).
- **Maps to field:**
 - EXPENSECATEGORY_FULL.NAME
 - EXPENSECATEGORY_U.NAME
- **Use case:** Run a report filtered by assignments that use a certain Expense Category associated with them.

Parameter: Funding Source

- **Identifier:** p_funding_source
- **Details:** Enables a user to select a Funding Source.
- **P6:** Provides a hierarchical list filled with Funding Sources where a user has access.
- **Return value:** The short name of the Funding Source (unique).
- **Maps to field:**
 - FUNDINGSOURCE_FULL.NAME
 - FUNDINGSOURCE_U.NAME
- **Use case:** Filter the report data to include only projects that have the selected Funding Source assigned.

Parameter: Portfolio ID

- **Identifier:** p_portfolio_id
- **Details:** Enables a user to select a Portfolio where the user has access.
- **P6:** Provides a list of Portfolios where the user has access.
- **Return value:** The portfolio short name (unique).
- **Maps to field:** PROJECTPORTFOLIO_FULL.NAME
- **Use case:** Filter the report data to include only the projects in a Portfolio.

Note: If there is a user portfolio that shares the same name as one of the global portfolios, then the report will return data for both when you run it. You'll need to use the Portfolio Name and User ID to make a unique constraint.

Parameter: Project ID

- **Identifier:** p_project_id
- **Details:** Enables a user to select one or more projects where the user has access.
- **P6:** Click the Projects _ menu, and select **Open Project** to display the Project list. Enables switching between Template and Regular projects.
- **Return value:** The project short name (unique).
- **Maps to field:** PROJECT_FULL.ID

- ▶ **Use case:** Run a report where the data comes from selected projects.

Parameter: Project Code Value

- ▶ **Identifier:** p_p_code_val__<short name>
- ▶ **Details:** Select a Project Code value. Note that underscores (_) are the only special character allowed. Do not use other special characters.
- ▶ **P6:** Provides a list containing the project code values for the Project Code whose short name matches the second part of the parameter. For example: If the short name was Scope, and there were four values – Local, Regional, Country, and Global – the list would display Local, Regional, Country, and Global in the list.
- ▶ **Return value:** Activity code value short name (unique per code type).
- ▶ **Maps to field:** PROJECTCODE_FULL.CODEVALUE
- ▶ **Use case:** Filter the set of projects loaded to those projects that have the user-selected Project Code Value assigned to them.

Parameter: User Defined Project Code

- ▶ **Identifier:** p_project_code_value"."<number> p_project_code_type"."<number>
- ▶ **Details:** Similar to the User Defined Activity Code, this parameter consists of two parameters in BI Publisher: One parameter returns the selected Project Code Value, and the other parameter returns the Project Code type ID. For each parameter you use, you must add both **p_project_code_value.#** and **p_project_code_type.#**. There must be a pair of numbers for the editor to work properly. You can have multiple sets to allow for more than one User Defined Code Value.
- ▶ **P6:** Provides a list populated with all the global Project Codes as the first level. Expanding a Project Code type will list all the values for the type. In the parameter table, only one row will represent both parameters. After you select a Project Code Value, both parameters will be set.
- ▶ **Return value:** The short name for p_project_code_value and the object id for p_project_code_type.
- ▶ **Maps to field:**
 - p_project_code_value maps to PROJECTCODE_FULL.CODEVALUE
 - p_project_code_type maps to PROJECTCODE_FULL.CODETYPEOBJECTID
- ▶ **Use case:** Create a report that enables the projects to filter based on a Project Code that the user defines. Unlike the Project Code Value parameter, the report creator should not hard code the Project Code type. Instead, they should write the query to enable the p_project_code_type.1 parameter to determine the Project Code type. This lets a report be more flexible in the filter criteria.

Parameter: User Defined Resource Code

- ▶ **Identifier:** p_resource_code_value "." <number> p_resource_code_type "." <number>

- ▶ **Details:** Similar to the other user-defined codes this parameter consists of two parameters on the report in BI Publisher: One parameter returns the selected Resource Code Value, and the other parameter returns the Resource Code type ID. For each parameter you use, you must add both **p_resource_code_value.#** and **p_resource_code_type.#** There must be a pair of numbers in order for the editor to work properly. You can have multiple sets to allow for more than one user-defined code value.
- ▶ **P6:** Provides a list populated with the Resource Codes as the first level. Expanding a Resource Code type will list all the values for the type. In the parameter table, only one row will represent both parameters. After you select a Resource Code Value, both parameters will be set.
- ▶ **Return value:**
 - p_resource_code_value: short name for the code value
 - p_resource_code_type: object id for the code type
- ▶ **Maps to field:**
 - p_resource_code_value maps to RESOURCECODE_FULL.CODEVALUE
 - p_resource_code_type maps to RESOURCE_CODE_FULL.CODETYPEOBJECTID
- ▶ **Use case:** Create a report that allows for the resources to be filtered based on a Resource Code Value and type that the user defines. Unlike the Resource Code Value parameter, the report creator should not hard code the resource code type. Instead, they should write the query to enable the p_resource_code_type.1 parameter to determine the resource code type. This allows a report to be more flexible in the filter criteria.

Parameter: Resource Code Value

- ▶ **Identifier:** p_r_code_val__<short name>
- ▶ **Details:** Select a Resource Code value to use for filtering loaded resources for a report.
- ▶ **P6:** Provides a list populated with the resource code values for the resource code type indicated by the <short name> at the end of the parameter. For example: If the short name was Department, the list would display Engineering, Marketing, Research, and Development as the values in the list.
- ▶ **Return value:** The short name of the Resource Code Value (unique per code type).
- ▶ **Maps to field:** RESOURCECODE_FULL.CODEVALUE
- ▶ **Use case:** Create a report where the resources filter based on the resources that match the user-selected Resource Code Value for a particular Resource Code type. The report data query must hard code the Resource Code type for matching resources to the returned code value.

Parameter: Resource Team

- ▶ **Identifier:** p_resource_team
- ▶ **Details:** Enables resource team selection.
- ▶ **P6:** Provides a list populated with resource teams that the user has access to view.
- ▶ **Return value:** The short name of the resource team (unique).
- ▶ **Maps to field:** RESOURCETEAM_FULL.NAME

- **Use case:** Filter a report to load resources that are on the selected resource team.

Parameter: Resource ID

- **Identifier:** p_resource_id
- **Details:** Select a resource to filter a report.
- **P6:** Provides a list populated with resources that the user has access to view.
- **Return value:** The short name of the resource (unique).
- **Maps to field:** RESOURCES_FULL.NAME
- **Use case:** Filter the activities in a report based on the user-selected resource assigned to the activity.

Parameter: Responsible Manager

- **Identifier:** p_responsible_manager
- **Details:** Select a responsible manager (OBS).
- **P6 GUI:** Provides a list populated with the OBS structure that the user has access to view.
- **Return value:** The short name of the OBS (unique).
- **Maps to field:** PROJECT_FULL.OBSNAME and OBS_FULL.NAME
- **Use case:** Filter a report to load only the projects that have the user-selected responsible manager.

Parameter: Risk Category

- **Identifier:** p_risk_category
- **Details:** Select a Risk Category.
- **P6:** Provides a list populated with all the Risk Categories that the user has access to view.
- **Return value:** The name of the Risk Category (unique).
- **Maps to field:** RISK_FULL.RISKTYPE
- **Use case:** Filter a report to load only the Risks of the user-selected category.

Parameter: Role Team

- **Identifier:** p_role_team
- **Details:** Select a Role Team.
- **P6:** Provides a list populated with all the Role Teams the user has access to view.
- **Return value:** The name of the Role Team (unique).
- **Maps to field:** ROLLTEAM_FULL.NAME
- **Use case:** Filter a report of Resources to include only the Resources that are assigned to the user-selected Role Team.

Parameter: Role

- **Identifier:** p_p6_role_id
- **Details:** Select a Role.
- **P6:** Provides a list populated with all the Roles the user has access to view.

- **Return value:** The short name of the Role (unique).
- **Maps to field:** ROLL_FULL.ID
- **Use case:** Filter a report of Resources to include only the Resources that have the user-selected Role.

Parameter: Timesheet Period

- **Identifier:** p_timesheet_period_start and p_timesheet_period_end
- **Details:** This parameter consists of two parameters on the report in BI Publisher, but is represented by a single row in parameter list for P6. The user will select a timesheet period, and it will set the Start Date to p_timesheet_period_start and the End Date to p_timesheet_period_end.
- **P6:** Drop down menu of the Timesheet Periods in the database. Selecting a Timesheet Period will return the Start Date and End Date in the parameters.
- **Return value:** The Start Date and End Date of the Timesheet Period selected by the user.
- **Maps to field:** N/A
- **Use case:** Filter a report to look for a date between two dates of a Timesheet Period. Instead of adding two date parameters and making the user manually enter the Start and End of the period, the user can use a drop down with the Timesheet Periods in the database.

Parameter: User

- **Identifier:** p_p6_user_id
- **Details:** Select a P6 EPPM user name.
- **P6:** Provides a list populated with the P6 users that the logged-in user has access to view.
- **Return value:** The user name (unique).
- **Maps to field:** USERS_FULL.NAME

Primitive Parameters

In BI Publisher, when you create a parameter, a field Data Type allows the following options: String, Integer, Boolean, Float. The default type is String, and you should use it for all supported enumeration and dynamic parameters. For primitive parameters, select the appropriate option for the parameter that you are configuring.

Type: Date

- **Editor:** P6 will use the date picker to select the date.
- **Return:** The date string in the format entered in BI Publisher.

Type: Boolean

- **Editor:** Will use a check box editor.
- **Return:** Either true (selected) or false (cleared).

Type: Integer

- ▶ **Editor:** Will use a default text box without validation.
- ▶ **Return:** The text entered in the box.

Type: Float

- ▶ **Editor:** Will use a default text box without validation.
- ▶ **Return:** The text entered in the box.

Allowing for Multiple Values Returned via a List

If you need a supported parameter to return a comma separated list of values for an IN clause in the data template, do the following:

- 1) In BI Publisher, set the parameter type to **Menu**. This will allow you to link it to a list.
- 2) Create a hard-coded list or a query to get the possible values for the parameter.
See examples of this in BI Publisher's pre-packaged reports.
- 3) Select the list of values you just created for the parameter.
- 4) Check the **Multiple Selection** option for the parameter.

If a parameter allows for multiple selection, P6 allows selection of multiple values from the list or allows users to continue to assign values without closing the list. When the return value appears, parameter values are comma-delimited.

Reporting in Primavera Unifier

The Oracle Business Intelligence Publisher (BI Publisher) integrates with Unifier Cloud Service to deliver on-demand web-based reporting.

The following sections describe how to configure Oracle Business Intelligence Publisher (BI Publisher) for Primavera Unifier Cloud Service.

Creating a BI Publisher Report in Primavera Unifier

To create a new BI Publisher report:

- 1) Sign in to your Unifier environment.
- 2) Click the **Company Workspace** tab.
- 3) In the left pane:
 - a. Switch to **Administration Mode**.
 - b. Expand **Configuration**.
 - c. Select **Custom Prints and Reports**.
- 4) In the right pane:
 - a. Select the **New** (dropdown arrow).
 - b. Choose **Report with Internal Data Model** or **Report with External Data Model**.

If you choose Report with External Data Model, you will continue using the steps in the section.

Notes:

- **Internal data model:** Unifier maintains the data model file internally. Any changes in the data views, data links, query parameters, and additional parameters are being applied and saved in the data model when the BIP report is saved.
 - **External data model:** Users need to upload the data model (.xdm) file and maintain it externally. This means that the user may modify the .XDM file manually or by using the BIP data model editor.
-

- 5) In the **General** tab of the **Custom Report** dialog box:
- a. In the **Name** field, enter a name for the report.
 - b. In the **Description** field, enter a brief description about the report (maximum 250 characters and non-ASCII characters are permitted).
 - c. The **Data Model Type** field is a pre-populated and read-only field with values that can be internal or external based on the option selected at the time of creating the custom print.
 - d. In the **Report Level** field, select a type for the report.
 - e. In the **Main View** field, select a type of view for the report.
-

Note: The administrator can create user-defined views. Refer to the *Unifier Administration Guide* for more information on creating data views.

- 6) In the **Views** tab, click **Add** or **Remove** to edit the views for the report.
- 7) Click **Apply**.
- 8) When the **Sample Data** tab opens, you can download the sample XML data for creating or modifying report layouts:
- a. If you are using the Internal Data Model:
 1. Select a project/shell.
 2. Select a number of rows to return.
 3. Click **Generate**.
 4. Click **Download**.
 5. Save the XML file.
 - b. If you are using the External Data Model:
 1. Select a project/shell.
 2. Click **Generate**.
 3. Click **Download**.
 4. Save the XML file.

Note: You cannot download sample data until after you define and publish the report.

If you haven't already installed the BI Publisher plug-in for Microsoft Office, go to <http://www.oracle.com/technetwork/middleware/bi-publisher/downloads/index.html> and install one of the following:

- ▶ Oracle BI Publisher Desktop for 32-bit Office on Windows
- ▶ Oracle BI Publisher Desktop for 64-bit Office on Windows

Note: For the full list of system requirements, applications, and application version levels refer to the *Primavera Unifier Tested Configurations* in the Primavera Unifier Documentation Library.

Refer to the documentation included with BI Publisher for more information on using this plug-in.

- 1) Create a report layout in Microsoft Word to have templates available in Microsoft Word when you publish reports:
 - a. Open **Microsoft Word**.
 - b. Create an **RTF** file.
 - c. In the **BI Publisher** tab:
 1. Select **Sample XML** in the **Load Data** section.
 2. Upload the XML file you created above.
 3. Create a table for the layout using the Table Wizard included with the BI Publisher plug-in.
 4. Save and close the RTF file.
- 2) Return to the **Custom Report** dialog box.
- 3) In the **Template File** tab (consisting of Add, Modify, Remove, and Download options)
 - a. Click **Add** to open the template window.
 - b. Enter information in the following fields: Template Name, Template Type, and Report Layout File.
 - To localize the Custom Report output for different languages, you can provide XLIFF files for RTF-type templates here.
 - The Template Name field accepts spaces and other characters.
 - c. Click **Browse** to select the template file for uploading.
 - d. Click **OK** to confirm the upload.
- 4) Return to the **Company Workspace** tab.
- 5) In the left pane:
 - a. Expand **Configuration**.
 - b. Select **Custom Reports**.
- 6) In the right pane, select the report you want to publish.
- 7) Click the **Status** dropdown on the toolbar and select **Published**.

Making a New Report Appear in Primavera Unifier

To make newly defined reports appear:

- 1) Go to the **Company Workspace** tab and switch to the Administrator mode.
- 2) Navigate to **Configuration** (User Mode Navigator).

Note: If the company does not use User Mode Navigator, you will be able to the report name under Access Control of Project or Program.

- 3) Open the **project/shell User Mode Navigator**.
- 4) In the dialog box, find the newly created BIP reports on the right side and move them to the left side under **Reports->Custom**.
- 5) Save the changes and close the popup window.
- 6) Select the **project/shell User Mode Navigator** and click the **Deploy** button to ensure the changes take effect.

Note: BI Publisher User Group requirement: To be able to create reports, Users must be added to BI Publisher Author user group through OIM.

Assigning Permissions for Primavera Unifier Reports

Complete the following steps to assign permissions for reporting:

- 1) Log in to Primavera Analytics.
- 2) Click **Catalog**.
- 3) Under **Shared Folders**, highlight the **UnifierReports** folder.
- 4) In the **Tasks** pane, click **Permissions**.
- 5) Select the following checkboxes:
 - ▶ **Apply Permissions to sub-folders**
 - ▶ **Apply permissions to items within this folder**
- 6) Set permissions for the following roles:
 - ▶ **BI Consumer: Open**
 - ▶ **BI Content Author: Traverse**
 - ▶ **BI Service Administrator: Full Control**
- 7) Click **OK**.

Creating a Data View in Primavera Unifier

You will create data views to operate as data sources for custom reports.

To create a data view:

Note: Refer to the *Unifier Administration Guide* for more information.

- 1) Sign in to your Unifier environment.
- 2) Click the **Company Workspace** tab.
- 3) In the left pane:
 - a. Switch to **Administration Mode**.
 - b. Select **Data Structure Setup > Data Views**.
- 4) In the **User-Defined Data Views** window, click **New**.
- 5) Enter the information needed for your data view. See **Tips** for more information.
- 6) Click **Apply** and **OK**.
- 7) Highlight the new view and select **Status, Published**.

Tips

The SQL definition will consist of the SQL query used to populate the report with specific data based on the desired output.

Note: If using the data view as a Main view in a Project level report, you must have a column named **project_id**. If using the data view in a Program level report, you must have a column named **program_id**.

For example, a Project level data view for a company registry "companyReg" can be like this below:

```
SELECT    pu.PID as PROJECT_ID,
          p.PROJECTNUMBER as PROJECTNUMBER,
          c.COMPANYNAME as COMPANY_NAME,
          c.SHORTNAME as COMPANY_SHORT_NAME,
          c.DESCRPTION as COMPANY_DESCRIPTION,
          c.CONTACTNAME as COMPANY_CONTACT_NAME,
          u.FULLNAME as USER_FULL_NAME,
          u.USERID as USER_ID,
          u.USERNAME as USER_NAME,
          u.EMAIL as USER_EMAIL
FROM      companyReg_SYS_USER_INFO_VIEW u,
          companyReg_SYS_COMPANY_INFO c,
          companyReg_SYS_PROJECT_INFO p,
          companyReg_SYS_PROJECTUSER pu
WHERE     c.COMPANYID = p.COMPANYID
          AND p.PID = pu.PID
```

AND u.userid = pu.userid

Creating a Custom Report in Primavera Unifier

Unifier integrates with BI Publisher to deliver on-demand web-based reporting through Unifier.

Custom Reports, built in BI Publisher, enable the Company Administrator (or power user) to build visually stunning, detailed reports. For example:

- ▶ A report that combines information from multiple (and possibly non-linked) Business Processes (BPs)
- ▶ A report that needs professional looking graphics, charts, images, or clip art

The following sections explain the steps you need to take for creating Custom Reports in Unifier with BI Publisher.

Download and Install BI Publisher Desktop for Microsoft Office

To download and install BI Publisher desktop for Microsoft Office:

- 1) Download **Oracle BI Publisher Desktop for Microsoft Office** from:
<http://www.oracle.com/technetwork/middleware/bi-publisher/downloads/index.html>
Ensure that you choose the BI Publisher Desktop version (32bit vs. 64bit) based on your version of Microsoft Office 32 bit or 64 bit.

Tip: If your Microsoft Word executable is found at C:\Program Files (x86)\Microsoft Office\Office12\WINWORD.EXE, then you have the 32 bit version).

- 2) Run the installer.
No additional setup is required after running the installer. To ensure that the installation was successful, check to see that the plug-in has been added to your Microsoft Word user interface. The BI Publisher ribbon appears after launching Microsoft Word.

(Optional) SQL Text Editor

Once you start to write more complex reports, having a text editor that highlights SQL keywords can make a big difference. For your needs, a simple editor such as Notepad++ would suffice. Ensure that you set the Language to SQL. You can copy the queries developed through the text editor and paste them into the Data View window.

(Optional) SQL Developer

You can download Oracle SQL Developer from:
<http://www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html>

- 1) Download the first option in the list: Windows 64-bit - zip file includes the JDK 7
- 2) Extract the file onto your computer (for example, C:\Oracle\sqldeveloper\)
- 3) Run **sqldeveloper.exe**

Review Unifier Database Table Structure

Before building your SQL statement, review the Unifier database table structure.

You can find the Unifier database table name in two ways:

1) Using Entity Reference (ER) Views

1. Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **Data Structure Setup > ER Views**.
2. Open **Business Processes View**.
3. Use your browser search function and search for a BP you want to report against (for example, Invoices).
4. The name of the table appears under the BP Name, in blue (for example, UNIFIER_UI). The word "unifier" is the prefix and common to all BP tables. The UI is the ID of the Invoices BP).

2) Using BP IDs

1. Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **uDesigner > Business Processes**.
 2. Locate the BPs that you want to use in the report and find the IDs (first column).
 3. Add your deployment prefix and add an underscore (_) to get the table name (for example, UNIFIER_UI).
- Note:** To access the Line Items of a BP, add `_LINEITEM` to the table name (for example, UNIFIER_UI_LINEITEM).

The following lists the column names:

- ▶ Field Name
- ▶ Definition
- ▶ SQL Type
- ▶ Label
- ▶ Description

Connect to Unifier Database

A direct connections to the Unifier database using SQL Developer is not possible. The creation of SQL queries must be done in Unifier Data Views, or by pulling rows down from Unifier into a local database.

If you need to pull the data into a local database for the purpose of SQL development, Oracle offers Oracle Database 11g Express Edition (XE) free of charge. You can install this lightweight DB on your computer and use it for development purposes.

Download Oracle Database Express Edition (XE). Refer to the XE documentation for instructions on how to install the software and create a local database.

(Optional) Pulling Down the Data

In Unifier, once you know the table names, you can export 200 lines so it can be inserted into your local Database (DB). To pull down the data:

- 1) Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **Data Structure Setup > Data Views**.
- 2) Click **New** to create a new Data View.
- 3) Enter a name (for example, Export DV) and a Label (for example, Export Data View).
- 4) In the SQL Definition field, enter the following for a particular table: `SELECT * FROM <tablename>`
- 5) Click **OK**.
- 6) Click the **Status** drop-down list and set the newly created Data View to **Published**.
- 7) Highlight the Data View you just created and click **Data** on the toolbar.
- 8) In the window, click **Export as SQL** on the toolbar to save the **.sql** file locally. This SQL file includes the CREATE and INSERT commands required to get your data into your local XE database.

At this point, you can create the table and insert the exported rows into your local XE database using tools like SQL Developer.

Repeat the steps above for any tables that you like to access offline.

Note: You can reuse the same Data View multiple times; however, you need to do find and replace in the downloaded SQL to ensure that the table names match the names that are in Unifier.

Since you have a subset of the database locally, you can use SQL Developer to write complex queries, offline, before bringing them into Unifier.

Writing Your First Statement

To write your first statement:

- 1) Create a new SQL file in your text editor of choice.
- 2) Write the following SQL statement: `SELECT * FROM unifier_ui`
- 3) To manage the amount of data generated, select only the columns you want to include in your report.
 - ▶ See **Review Unifier Database Table Structure** (on page 35) for a list of all column names.
 - ▶ Instead of using the asterisk, or star, (*) operator, specify which fields you want to include in our query result.
- 4) Replace the asterisk, or star, (*) operator in your statement with a few column names (for example, `SELECT project_id, ID, record_no, title, status, UVEVENDORNAMETB50, amount FROM unifier_ui`)
- 5) Assign an alias to your table to limit the typing that you need to do.
Example

Name your Invoice table "inv" by adding "inv" after the table name. You can then refer to each column using the alias, instead of the full name. Your statement should look like the following:

Note: The Query Builder re-formats the layout of your SQL.

```
SELECT inv.PROJECT_ID,
       inv.ID,
       inv.RECORD_NO,
       inv.TITLE,
       inv.STATUS,
       inv.UVEVENDORNAMETB50,
       inv.AMOUNT
FROM unifier_ui inv
```

- 6) Add aliases to the column names to make the column names easier to understand when building the BI Publisher report

Example

```
SELECT inv.PROJECT_ID,
       inv.ID AS inv_ID,
       inv.RECORD_NO AS inv_record_no,
       inv.TITLE AS inv_title,
       inv.STATUS AS inv_status,
       inv.UVEVENDORNAMETB50 AS vendor,
       inv.AMOUNT AS inv_amount
FROM unifier_ui inv
```

- 7) (Optional) Choose any other columns from the invoice table that you want to include in your report.

Note: Unifier requires a field called "PROJECT_ID" since Unifier uses the field for data filtering (for example, Only the Invoices from the project that you are running the report from are shown). Always leave a field called PROJECT_ID in your main Data View to prevent receiving an error message when creating the Custom Report.

SQL JOIN

The following scenario describes how to use JOIN to combine rows from two or more tables based on a common column between them.

Scenario

Information is missing in your Invoice report and you cannot find the information in the Invoices table. Furthermore, you have a column called CREATOR_ID, but the column contains a number pointing to an ID in another table where all the User information is stored. You also have a column called PROJECT_ID which points to another table where all the Project Information is stored.

Solution

To find the table, follow these steps:

Note: You can use other SQL Joins to conduct the following.

Use a LEFT JOIN since this option returns all the rows from the left table (for example, Invoices) even if there are no matches in the right table (for example, User info table, Project info table, etc.).

To add the Invoice creator's name, match your CREATOR_ID column from the Invoices table to your Users table (unifier_sys_user_info_view). You can do so by using a LEFT JOIN on the USERID column of the User table. Assign an alias (sysuser) to add the information to the end of your statement:

Example

```
LEFT JOIN unifier_sys_user_info_view sysuser
ON (inv.CREATOR_ID = sysuser.USERID)
```

Look at the ER Views (System Models View) to see what options are available in your sysuser table.

Grab the following columns from the table: sysuser.FULLNAME, sysuser.EMAIL, and sysuser.WORKPHONE.

Add the columns to the end of the SELECT section of your query. Each column specified in the SELECT portion must have a comma after it, except for the last one.

Access the information about the Project or Shell (name, number, start/end dates, etc.). You can find this information in the table unifier_us_p, with the column proj.PID that is used to join the two tables.

Example

```
LEFT JOIN unifier_us_p proj
ON (inv.PROJECT_ID = proj.PID)
```

Grab the following columns from the table. and add the columns to the end of the SELECT section, before the FROM, of your query. Each column specified in the SELECT portion must have a comma after it, except for the last one.

Example

```
proj.UGENPROJECTNAME --> AS ProjectName,
proj.UGENPROJECTNUMBER --> AS ProjectNumber,
proj.UUU_PROJECT_START_DATE --> AS ProjectStartDate,
proj.UGENPROJENDDATEDO --> AS ProjectEndDate,
proj.UGENINITBUDGETCA --> AS InitialBudget,
proj.UGENREVBUDGETCA --> AS RevisedBudget
```

Ensure that you see the following codes:

Example

```
SELECT inv.PROJECT_ID,
       inv.ID AS inv_ID,
       inv.RECORD_NO AS inv_record_no,
       inv.TITLE AS inv_title,
       inv.STATUS AS inv_status,
       inv.UVEVENDORNAMETB50 AS vendor,
```

```
inv.AMOUNT AS inv_amount,  
sysuser.FULLNAME,  
sysuser.EMAIL,  
sysuser.WORKPHONE,  
proj.UGENPROJECTNAME --> AS ProjectName,  
proj.UGENPROJECTNUMBER --> AS ProjectNumber,  
proj.UUU_PROJECT_START_DATE --> AS ProjectStartDate,  
proj.UGENPROJENDDATEDO --> AS ProjectEndDate,  
proj.UGENINITBUDGETCA --> AS InitialBudget,  
proj.UGENREVBUDGETCA --> AS RevisedBudget  
FROM unifier_ui inv  
LEFT JOIN unifier_sys_user_info_view sysuser  
ON (inv.CREATOR_ID = sysuser.USERID)  
LEFT JOIN unifier_us_p proj  
ON (inv.PROJECT_ID = proj.PID)
```

SQL WHERE

The SQL WHERE Clause is used to filter records. Once you attain the columns that you want, you can filter the data on any column.

The following scenario describes how use WHERE to filter records.

Scenario

You only want to report on Invoices in an Approved or Pending state.

Solution

Use the following WHERE clause at the end of our statement: `WHERE inv.STATUS = 'Approved' OR inv.STATUS = 'Pending'`

Note: You can add multiple filters using the OR and AND operators.

SQL ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set by one or more columns.

You can sort the data in the SQL to save the time it takes for BI Publisher to generate the report.

Note: The BI Publisher can also sort the data.

You can use the ORDER BY operator.

Scenario

You want to sort your data first by Project ID and then by Invoice Record Number since that is the order the data will be reported on.

Solution

Add this to the end of our statement:

```
ORDER BY inv.PROJECT_ID,  
        inv.RECORD_NO
```

SQL Final Statement

When done with JOIN, WHERE, and ORDER BY, you can save your statement locally.

Example based on the scenarios presented in the previous topics:

Invoice Upper Form Data	SELECT inv.PROJECT_ID,		
	inv.ID AS inv_ID,		
	inv.RECORD_NO AS inv_record_no,		
	inv.TITLE AS inv_title,		
	inv.STATUS AS inv_status,		
	inv.UVEVENDORNAMET B50 AS vendor,		
	inv.AMOUNT AS inv_amount,		
Invoice Creator Data	sysuser.FULLNAME,		
	sysuser.EMAIL,		
	sysuser.WORKPHONE,		
Project Data	proj.UGENPROJECTNA ME	AS ProjectN ame,	
	proj.UGENPROJECTNU MBER	AS ProjectN umber,	

	proj.UUU_PROJECT_START_DATE	AS ProjectStartDate,	
	proj.UGENPROJENDDATEDO	AS ProjectEndDate,	
	proj.UGENINITBUDGETCA	AS InitialBudget,	
	proj.UGENREVBUDGETCA	AS Revised Budget	
	FROM unifier_ui inv		Note: No comma (,) on the last line of the SELECT section.
	LEFT JOIN unifier_sys_user_info_view sysuser		List of tables we're getting data from, including the fields we're using to match up the tables.
	ON (inv.CREATOR_ID = sysuser.USERID)		
	LEFT JOIN unifier_us_p proj		
	ON (inv.PROJECT_ID = proj.PID)		
	WHERE inv.STATUS = 'Approved' OR inv.STATUS = 'Pending'		Filtering of the data
	ORDER BY inv.PROJECT_ID,		Sorting of the data
	inv.RECORD_NO		

Adding Line Items

When you gathered all the information that you need, from the Upper form and associated tables (see **Writing Your First Statement** (on page 36)), you need to add the Line Items to your query results (for example, Invoice Line Items). The Line Items are stored in the UNIFIER_UI_LINEITEM table.

Example

There are two ways to match the Invoice to the Invoice Line Items:

1) JOIN the Line Items DB Table

By using a LEFT JOIN since it returns all the rows from the left table (Invoices) even if there are no matches in the right table (Invoice Line Items).

2) Add Line Items

Add the Line Items as a second (separate) SQL statement and have Unifier combine the two as two separate data views. Invoice Line Items will be a Sub Report View to the Invoices Main Report.

JOIN the Line Items DB Table

This method of matching the Invoice to the Invoice Line Items generates duplicate data (by the number of Line Items in the Invoice) in BI Publisher. As a result, this method is not recommended.

Add Line Items

This method creates a new SQL query for the Invoice Line Items. This will result in a nested XML file, where all the Invoices Line Items are nested within their respective Invoice.

To add Line Items as a second (separate) SQL statement follow these steps:

1) Create a new SQL statement in your SQL Text Editor (you can save the new file as Invoice_LI.sql) and add the following to begin your SQL query: `SELECT * FROM unifier_ui_lineitem inv_li`

That is to say, you are going to select all the columns from the unifier_ui_lineitem table and give the table an alias of inv_li.

2) Specify which columns you want to include in your results.

```
SELECT inv_li.RECORD_ID,  
       inv_li.LI_NUM,  
       inv_li.SHORT_DESC,  
       inv_li.UUU_UNIT_PRICE,  
       inv_li.UUU_QUANTITY,  
       inv_li.AMOUNT  
FROM unifier_ui_lineitem inv_li
```

This is a Sub Report View in Unifier; therefore, you do not need to include a PROJECT_ID column. Since Unifier needs to match the Invoice Line Items to the Invoices, you need to include the field that Unifier can use for the JOIN. You must include the inv_li.RECORD_ID column. If you look at the two tables (and compare it to the data in Unifier), you notice that the RECORD_ID column in the line item table matches up with the ID column of the Invoice table.

- 3) Add aliases to these column names so they are easier to understand when you are building your BI Publisher report.

```
SELECT inv_li.RECORD_ID AS inv_li_record_no,
       inv_li.LI_NUM AS inv_li_no,
       inv_li.SHORT_DESC AS inv_li_desc,
       inv_li.UUU_UNIT_PRICE AS inv_li_unit_price,
       inv_li.UUU_QUANTITY AS inv_li_quantity,
       inv_li.AMOUNT AS inv_li_amount
FROM unifier_ui_lineitem inv_li
```

SQL JOIN

Use this method when some data is not stored in the Invoices Line Item table. The following scenario describes how use JOIN to combine rows from two or more tables based on a common column between them.

Scenario

You need to find the "unit of measure" name for each line item.

Solution

This appears to be in the table SYS_DATA_OPTION.

Note: Replace "unifier" with your table prefix. See *Review Unifier Database Table Structure* (on page 35) for information about finding a table name.

```
LEFT JOIN SYS_DATA_OPTION sdo
ON inv_li.UGENUNITOFMEASUREPD = sdo.OPTION_VALUE
AND (sdo.DATA_NAME = 'Unit of Measure'
AND sdo.REGISTRY = 'unifier')
```

1. Grab the following column from the table: sdo.OPTION_NAME AS inv_li_uom and add them to your SELECT. The goal is to be able to access the cost code number and description for each line item, which can be found in the table unifier_budgetitem:

```
LEFT JOIN unifier_budgetitem bi
ON inv_li.BITEMID = bi.BITEMID
AND inv_li.BUDGETID = bi.BUDGETID
```

2. Grab the following columns from the table: bi.CODE & bi.ITEM and add them to your SELECT.

SQL ORDER BY Keyword

The ORDER BY keyword is used to sort the result-set by one or more columns.

You can sort the data in the SQL to save the time it takes for BI Publisher to generate the report. You can sort the data by Invoice (inv_li_record_no) and then by Invoice Line Item Number (inv_li.LI_NUM): ORDER BY inv_li_record_no, inv_li.LI_NUM

Note: Since Unifier nests the Line Items within the Invoices and in the processes making your table easier to read when troubleshooting, sorting by Invoice is not necessary.

SQL Final Statement

When done with JOIN and ORDER BY, you can save your statement locally.

Example based on the scenarios presented in the previous topics:

Example of a final Invoice Line Item SQL Statement:

```
SELECT inv_li.RECORD_ID AS inv_li_record_no,
       inv_li.LI_NUM AS inv_li_no,
       inv_li.SHORT_DESC AS inv_li_desc,
       inv_li.UUU_UNIT_PRICE AS inv_li_unit_price,
       inv_li.UUU_QUANTITY AS inv_li_quantity,
       inv_li.AMOUNT AS inv_li_amount,
       sdo.OPTION_NAME AS inv_li_uom,
       bi.CODE,
       bi.ITEM
FROM unifier_ui_lineitem inv_li
LEFT JOIN SYS_DATA_OPTION sdo
ON inv_li.UGENUNITOFMEASUREPD = sdo.OPTION_VALUE
AND (sdo.DATA_NAME = 'Unit of Measure'
AND sdo.REGISTRY = 'unifier')
LEFT JOIN unifier_budgetitem bi
ON inv_li.BITEMID = bi.BITEMID
AND inv_li.BUDGETID = bi.BUDGETID
ORDER BY inv_li_record_no, inv_li.LI_NUM
```

Creating a New Custom Report

To create a new custom report:

- 1) Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **Configuration** > **Custom Reports**.
- 2) Click **New** to create a new Custom Report (With Internal Data Model).
- 3) Enter a name (for example, Invoices) and a description.
- 4) For Report Level, choose **Project** since this report will be run at the Project level.

- 5) For Main View, select the Data View that you created.

Note: An error message appears if you do not add PROJECT_ID as one of the columns in your SQL statement. You can always go back to the Data View, mark the PROJECT_ID as Draft, correct the SQL statement, and re-publish.

- 6) Click the **View** tab.
 7) Click **Add**, on top, to add a new view.
 8) For Main View, select the Data View that you created.
 9) Enter a name and a tag for your data set (for example, Inv for Name and Inv for tag. The same name for both fields.).

For the Line Items view:

- 1) Click **Add**, on top, to add a new view.
 2) For Main View, specify the View Type as a Sub Report View.
 3) Enter a name and a tag for your data set (for example, Inv for Name and Inv for tag. The same name for both fields.).

Row	View Name	View Type	Data Set Name	Data Set Tag
0	Invoices	Main View	inv	inv
1	InvoicesLI	Sub Report View	LineItem	LineItem

At this point, set the relationship between the Main View and the Sub Report View that you have added.

- 1) Go to the **Data Links** section and click **Add**.
 2) Link the ID field, from the Invoices Data View, to the RECORD_ID field, from the Invoices Line Item Data View.

Example

Source Data Set: inv
 Source Element Name: INV_ID
 Target Data Set: LineItem
 Target Element: INV_LI_RECORD_NO

- 3) Click **OK** to add to add the link.
 4) Click **Apply**.

Building Report Template

Before you begin building a report template, you must have completed the steps in the preceding sections: SQL statement: **Writing Your First Statement** (on page 36) > Creating Data Views, Building Custom Report: Building Report Template

You can build a Rich Text Format (RTF) template and upload the template to your Custom Report in Unifier, and generate live reports.

Using Invoice as an example, the following explains how to build a simple RTF template (non-tabular report).

To create a simple RTF template:

- 1) Open Microsoft **Word**.

The application must have the Microsoft Word BI Publisher plug-in installed. See **Download and Install BI Publisher Desktop for Microsoft Office** (on page 34).

- 2) Click the **Word BI Publisher** ribbon.
- 3) Click **Sample XML** to import your sample data and wait until the data is loaded successfully.

Note: Alternatively, you can use a template file (for example, A Word template file from your customer).

- 4) Click **Repeating Group** to generate a loop on each Invoice. The Repeating Group window opens.
- 5) In the **For Each** field, enter a value in the Data Set Name for each of the Invoices that you chose, when defining your Custom Report (for example, inv). This does not apply to the Invoices Line Items.
- 6) In the **Group By** field, select a field for your Invoice loop. Use the `INV_ID` because it is a unique identifier for each Invoice.
- 7) (Optional) Insert a page break after each Invoice to keep your report formatted.
- 8) (Optional) Select the Data already sorted option. You can select this option because you have already sorted your data in your SQL statement, using: `ORDER BY`.
- 9) Click **OK**.

The BI Publisher plug-in application adds a code to the document which includes: a start (for-each), a page break (page break), and an end (end) for your Invoice loop.

- 10) Add a blank line after "for-each" operator to make room for your Invoice information.
- 11) Click the field icon (**ab|Field**) to open the field browser.

The field browser window enables you to add fields from you XML sample data file.

- 12) After each "for-each" operator, double-click **Inv_Record_No** in the field browser.
- 13) Add a hyphen (–) [n-dash] after the record number.
- 14) Double-click **Inv_Title** to add the Invoice title.

Example

```
for-eachINV_RECORD_NO-INV_TITLE
```

```
page breakend
```

- 15) (Optional) Run the report by clicking PDF on the Word BI Publisher Ribbon (Word will prompt you to save your RTF file if you have not, already).

The first report is now ready.

The report only has the Invoice record number and title for each Invoice in your sample XML (the approved and pending ones because: `WHERE inv.STATUS = 'Approved' OR inv.STATUS = 'Pending'` in our SQL statement).

You can continue to build on the generated report by adding additional fields from the Upper Form of the Invoices BP, as well as headers, footers, formatting, and so forth.

Creating a Report with Line Items

You can use the Table Wizard to create a report with all the Line Items.

Using Invoice as an example, the following explains how to create a report with Line Items:

- 1) Add a blank line after `for-eachINV_RECORD_NO - INV_TITLE` operator and leave your cursor.
- 2) Click **Table Wizard** on the Word BI Publisher Ribbon.
- 3) Select **Table** as your Report Format and click **Next**.
- 4) Select **/DATA_DS/Inv/LinItem** as your Data Set (or the name that you had given your Line Item data set).
- 5) Select the fields that you want to add to the table. Since this is for the Invoice line items, you must only add fields that are specific to the line items.

If you need to add a field for "group on" (for example, `Inv_Li_Record_No`), the grouping process is similar to the process explained in the preceding section. Since you have grouped the records by Invoice in the preceding section, you can leave the value blank.

The order of the selected columns must match the order that you want in your table, except for the column that you use for grouping, which is outside the table. In the "Which fields do you want to show in your report" window, match the order as shown here:

```
Inv Li Record No --> In Li Record
Inv Li No --> Inv Li No
Inv Li Desc --> Inv Li Desc
Inv Li Unit Price --> Inv Li Unit Price
Inv Li Quality --> Inv Li Quality
Inv Li Amount --> Inv Li Amount
Inv Li Uom --> Li Uom
Code --> CodeInv
Item --> Item
```

- 6) In the "How would you like to group your report" window, leave every option as is because you have already grouped your data by Invoice in the preceding section.
- 7) In the "Which fields would you like to user to sort the data" window, within each table, sort the lines by `Inv_Li_No` (Invoice Line Item Number). Specify that this is a Number so that it is sorted correctly.
- 8) Click **Finish**.

The Table Wizard inserts the table and the necessary code:

```
for-eachINV_RECORD_No - INV_TITLE
```

Inv Li No	Inv Li Desc	Code	Item	Inv Li Quantity	Inv Li Uom	Inv Li Unit Price	Inv Li Amount
F INV_LI_NO	INV_LI_DESC	CODE	ITEM	INV_LI_QUANTITY	INV_LI_UOM	INV_LI_UNIT_PRICE	INV_LI_AMOUNT

page breakend

You must run the report to see what information is generated. While the data is correct, you need to work on formatting the data. See **Formatting Data** (on page 48) for details.

Formatting Data

To format the data generated:

- ▶ Provide a descriptive text (not SQL column names) for Column titles.
- ▶ Adjust the Column widths.
- ▶ Apply general table coloring (borders and shading, font sizes, cell alignment, etc.).
- ▶ Ensure that the "dollar" format is used for the two price columns (for example, \$110.00). See the details that follow.
- ▶ Add useful information, from the Upper Form of the Invoice, above the table. See the details that follow.
- ▶ Provide a "Total" for the amount column. See the details that follow.

Once finished, generate a PDF and repeat the process if necessary.

To change the formatting of the Price & Amount columns (for example, 110.0 > \$110.00):

- 1) Double-click on **INV_LI_UNIT_PRICE** (the code under the Unit Price field) to launch its BI Publisher properties.
- 2) Change the Formatting Type to **Number**.
- 3) Set the Formatting Format to **\$#,###0.00;(\$#,###0.00)** (paste in the blue text).
- 4) Repeat the preceding steps for the Amount column.

To add useful information, from the Upper Form of the Invoice, above the table:

Note: Tables are efficient formatting tool for organizing data from the Upper Form. Include separate columns for the field name (align right) and the field value (align left). You can also hide the borders if you prefer.

- 1) Create a 4-row and 2-column table and add field names.
- 2) Place your cursor where the first inserted field value must be entered.
- 3) Click the field icon (**ab|Field**) to open the field browser and add fields from your XML sample data.
- 4) Add useful information such as Creator, Email, and Status (or other fields you added to your SQL statement) above our Line Item Table.

- 5) Double-click on the correct field value from the Field window to add the Title to the report (for example, add Invoice title in the Title field).

To provide a "Total" for the amount column, using Invoice as an example:

- 1) Right-Click somewhere in the last row of your Invoice Line Item table and select **Insert > Insert Rows Below**.
- 2) Highlight all the columns in the new row, except for the last one, and **Merge** the cells (from the right-click menu).
- 3) Click in the newly created cell and enter: Total
- 4) Right-align the cell.
- 5) Place your cursor into your last column of the new row.
- 6) Click the field icon (**ab|Field**) to open the field browser and add fields from your XML sample data.
- 7) Click **Inv_Li_Amount** column to highlight.
- 8) Set the calculation (at the bottom) to **Sum**.
- 9) Click **Insert** to add the calculation into the table.
- 10) (Optional) Select the Total row and make the text bold.

Adding Summary Page to Report

Your report lists details. The Summary page contains a table with a summary of all the details (for example, Invoices) as well as charts to add graphical information.

To add a summary page to your report, using Invoice as an example:

- 1) Add Project information.

In preceding sections, you have set your template to loop through each Invoice. In order to create a summary page, you need to include the following information:

- a. Using Word, insert a page break before the "for-each code" at the top of the document. Do not use the page break command in BI Publisher.
- b. Add a title to the Summary page (for example, Invoice Report).
- c. Click the field icon (**ab|Field**) to open the field browser and add information about the project, if you have not done so in your SQL statement.
- d. Add additional information about the Project below the title (in a table) such as the Project number, Project name, Start/End dates and Initial/Revised budgets.
- e. Format dollar values as numbers with the formatting **\$#,##0.00;(\$#,##0.00)**. You can format dates as type Date using date formatting options such as: MM/dd/yyyy

- 2) Add Summary table.

A summary table contains the Invoice data that goes in the report. You can use the Table Wizard, similar to the process in the "Creating a Report with Line Items" section.

- a. Place the cursor on a new line under the table on the title page, where you want your table of Invoices to appear.
- b. Click **Table Wizard** on the Word BI Publisher Ribbon.
- c. Select **Table** as your Report Format and click **Next**.
- d. Select **/DATA_DS/Inv** as your Data Set (or the name you used in your Invoice data set).

- e. Select the fields that you want to add to the table. Since this is for the Invoice, you must only add fields that are specific to the Invoices and not the Invoices Line Items.
You do not need to group because your SQL statement provides one line per Invoice.
- f. Sort by Invoice Record Number, in the "Which fields would you like to use to sort the data?" window.
- g. Click **Finish**. The Table Wizard inserts the table and the necessary code.
- h. Format the data. See **Formatting Data** (on page 48) for details on how to format the data.

Adding a Chart

Using Invoice as an example, you can include a chart in the summary page to demonstrate how the Invoices are split.

There are several chart types available in BI Publisher. The following is for creating a Pie-chart.

To add a pie chart:

- 1) Place the cursor above the summary table. The goal is to include the chart between the Project summary table and Invoice summary table because the table can get long and expand to the next page.
- 2) Click **Chart** on the Word BI Publisher Ribbon.
- 3) Set the **Chart Type** (on the right) to **Pie Chart**.
- 4) (Optional) Select one of the Chart Styles.
- 5) Drag **Inv_Amount** from the Data tree to the Values box to ensure that the size of each pie slice is determined by the dollar amount of the Invoice.
- 6) Drag **Inv_Title** from the Data tree to the Labels box to ensure that the:
 - ▶ Pie slices are determined by Invoice
 - ▶ Invoice Title appears in the legend
- 7) (Optional) Click **Preview** (top right corner) to see a preview of the chart and make changes if necessary.
- 8) (Optional) Use the Properties table on the right side to set Chart Title, Legend properties, and so on.
- 9) Click **OK**. You change the chart settings by double-clicking the chart in the template.
- 10) (Optional) Add spacing between the chart and the two tables and center-align the chart.

Adding Headers and Footers

Use a 3-column table in the header and footer of your template to allow for a uniform adjustment of items such as titles, logos, and page numbers. Use Microsoft Help to learn how to add the first page to your document that does not include Header or Footer.

Adding Images from Unifier

Note: Microsoft Word does not support form fields in the header and footer. If need to add an image to the header or footer of your Custom Report and you want the image to repeat on each page, see the "Adding

BI Fields to the RTF Header or Footer" section.

To add an image (Examples: Company logo, Shell image, or image picker from a BP record: jpg, gif, or png) from Unifier into your Custom reports, use the Sample XML file (Sample Data) that you have created and exported into Unifier.

Note: Company logo cannot be displayed in Unifier interface. Create a company-level business process to place your Company logo.

The following shows the procedure by using an example:

- 1) Open the XML file.
- 2) On top, identify the XML elements that are blank (shown in **bold** in the following code).

```
<?xml version="1.0" encoding="UTF-8"?>
<DATA_DS>
<uuu_p_reportByF></uuu_p_reportByF>
<uuu_p_timeZoneF></uuu_p_timeZoneF>
<uuu_p_diffMinutesF>0</uuu_p_diffMinutesF>
<uuu_p_sysyTimeZoneID></uuu_p_sysyTimeZoneID>
<uuu_p_searchConditionF></uuu_p_searchConditionF>
<uuu_p_urlF></uuu_p_urlF>
<uuu_p_sessionIdF></uuu_p_sessionIdF>
<uuu_p_companyRegistryF></uuu_p_companyRegistryF>
<inv>
  <PROJECT_ID>1012</PROJECT_ID>
  <INV_ID>1</INV_ID>
  <INV_RECORD_NO>INV-001</INV_RECORD_NO>
  <INV_TITLE>Lumber Contract - Initial Invoice</INV_TITLE>
  <INV_STATUS>Approved</INV_STATUS>
```

Note: At runtime, these blank elements are fully populated with information about the Unifier server base URL, the User's session ID, and the Unifier company registry.

- 3) Using the included parameters, plus the ID of a specific image (the image that you want), construct a URL of the format:

```
<uuu_p_urlF>CompanyRegistry=<uuu_p_companyRegistryF>&sessionId=<uuu_p_sessionIdF>&id=<image_ID>
```

Note: To build a similar URL in your BI Publisher report, ensure that you have the Image ID because the other three parameters have already been identified.

- 4) Access the Image ID of the image that you want and:

Note: If applicable, you can find the Company logo Image ID in the companylogo column of the table sys_company_info.

- a. JOIN the information into your existing Data Views.
or
- b. Add the information as a new Data View, which you can add as a Sub-Report View to any Custom Report. The SQL to create a new Data View:

```
SELECT companylogo
FROM sys_company_info
WHERE companyregistry = 'unifier';
```

5) Define the variables:

Once you have an Image ID, you must define the variables needed to build the image URL by hiding the variables in a Data Field:

- a. Add a data field somewhere at the top of your report. This can be any field. You only need a placeholder for your variables.
- b. Double-click on the newly added data field and click the **Advanced** tab. Delete the text in the code box, if any.
- c. Add the following text into the code box:

```
<?variable@begin:uuu_p_urlF;(./uuu_p_urlF)[1]?>
<?variable@begin:uuu_p_companyRegistryF;(./uuu_p_companyRegistryF)
)[1]?>
<?variable@begin:uuu_p_sessionIdF;(./uuu_p_sessionIdF)[1]?>
<?variable@begin: image;( /DATA_DS/co/COMPANYLOGO)?>
```

Notes:

- If you are adding a Shell image or Image Picker Data Element, you do not need that last line. Add the last line only if you have created a Company log Data View (If applicable, you can find the Company logo Image ID in the companylogo column of the table sys_company_info).
 - The example code presumes that you added an unlinked Sub Report View with the Data Set Tag “co” (for Company). You can change the Data Set Tag “co” (for Company) to the value that you have in the last column of Company Logo row in the Views used as data sets table on the Views tab of the Custom Report dialog.
 - If you want to insert the Project Shell ID, you must access that variable. Depending on where you are in your grouping, the variable can be: <?variable@begin:image;(SHELLIMAGE)[1]?>
-

To access the Project or Shell Image ID:

- 1) Go to table: **unifier_shell_info**.
- 2) Identify the **shellimage** column of the table. The Project or Shell Image ID is in the shellimage column.

You can JOIN this table to your main report view based on the **unifier_shell_info.pid** column, which contains the project ID.

After you defined your variables, you can add an image. The BI Publisher leverages the Alternative Text of an image to dynamically set the image source. For more information, refer to the *BI Publisher Report Designer's Guide*.

To add a dummy image:

- 1) Create an image (for example, Dummy_Image.jpg) and size it appropriately (for example, 300 x 200 px).
- 2) Right-click on the image and select Size...
- 3) Click the Alt Text tab of the Size dialog
- 4) Paste the following into the Alternative Text box:
`url:{concat($uuu_p_urlF, 'companyRegistry=', $uuu_p_companyRegistryF, '&sessionId=', $uuu_p_sessionIdF, '&id=', $image)}`

Note: The last parameter (\$image) was set as a variable to either the Company logo or the Project or Shell image. You must adjust the parameter if you want to include both images. Ensure that you use separate variable names for each.

To test an image:

The three parameters that you used to build your URL to the image are not downloaded as part of the sample data:

```
uuu_p_urlF
uuu_p_sessionIdF
uuu_p_companyRegistry
```

You must upload your RTF template to Unifier, publish the Custom Report, and then run the report to ensure that your image was inserted successfully.

Adding BI Fields to RTF File Header or Footer

Microsoft Word does not support form fields in the header and footer. As a result, you need to populate the headers and footers with text or images from Unifier. Refer to the *BI Publisher Report Designer's Guide* for more details.

To populate the headers and footers with text or images from Unifier, follow these steps:

- 1) At the top of our RTF template (in the body, not the header), place the following tags:
`<?template:header?>`
`<?end header?>`
- 2) These tags form the start and end of our header. In between them, place the text and images want to include in the report header. Use a 3-column table in the header and footer of your template to allow for a uniform adjustment.
- 3) Double-click into the header of the document. Ensure that none of the BI Publisher fields are highlighted.

- 4) Add the text: <? call@:header?> to place everything in the tags added above into the header at report runtime.

Importing Template into Unifier

After you have completed creating your Template, you can upload your template into Unifier.

To upload your template into Unifier:

Note: Invoices is used as an example.

- 1) Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **Configuration** > **Custom Reports**.
- 2) Open your Invoices Custom Report.
- 3) In the **Report File** tab (consisting of Add, Modify, Remove, and Download options)
 - a. Click **Add** to open the template window.
 - b. Enter information in the following fields: Template Name, Template Type, and Report Layout File.
 - To localize the Custom Report output for different languages, you can provide XLIFF files for RTF-type templates here.
 - The Template Name field accepts spaces and other characters.
 - c. Click **Browse** to select the template file for uploading.
 - d. Click **OK** to confirm the upload.
- 4) Click **Browse** and upload the RTF file.
- 5) Click **OK** to close the window.
- 6) **Publish** your report by selecting your report from the log and selecting **Status** > **Published** from the toolbar.

Your report is added to list and Users can access the report from Projects, if they have the appropriate permissions.

Adding a Report to Navigator

To add your report to the Navigator:

- 1) Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **Configuration** > **Navigator** (User mode).
- 2) Open your Project or Shell Navigator. You must be able to see your Custom Report on the right-hand side. If you do not see your Custom Report ensure that you have published the report. See **Importing Template into Unifier** (on page 54).
- 3) Add your Custom Report to the Reports section on the left-hand side (in the Navigator, you can create a new subfolder under Reports called Advanced Reports).
- 4) Click **OK** to close the window.
- 5) Deploy the Navigator by highlighting it and clicking **Deploy** from the toolbar.

Setting Permissions on Report

You must give permissions to users so that they can run the new, custom report.

You can give permissions using a Project template (to distribute the permission to all projects), or do it on case-by-case bases, one project at a time.

To set permissions:

- 1) Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **Company Sponsored Shells > Projects**.
- 2) Find your Project (the project that you want to add the Custom Report to) and open.
- 3) Click **Access Control**.
- 4) Go to the Custom Report, click the report to open the Permission Settings window.
- 5) Add View access for any Users or Groups that you want to be able to run the Custom Report in your Project.

Repeat the preceding steps if you want to add your Custom Report to other Projects.

Running the Report

To run your Custom Report:

- 1) Navigate to you Project that has your Custom Report (for example, The Project with Invoice BPs created and in an approved or pending state)
- 2) Run the report.

Advanced BI Publisher Functions

The following topics explain the advanced functions of the BI Publisher.

Conditional Formatting

With BI Publisher, it is easy to use conditional formatting to highlight table cells or entire table rows using conditional formatting.

Example

Highlight invoices with an amount over \$10,000.

Highlighting a Table Cell

- 1) After you create a table, place the cursor in the cell where want to apply conditional formatting.

Note: The cell must be either text or a Data Field.

- 2) Click **Conditional Format**. The BI Publisher Properties window open on the Properties tab.
- 3) From the Data field drop-down list select the data element that you want to evaluate to determine the conditional highlighting. This does not need to be the same data field in the cell that you are trying to highlight.

- 4) Specify whether that Data Field is a Number or Date/Text Field.
- 5) Enter the conditions for this Data Field using the pertinent drop-down list.
- 6) For each condition, specify the formatting that you want to be applied when that condition is met.

Only two conditions can be entered by using this window. If you have more than two conditions, you can click on the Advanced tab and copy/paste the conditions already entered to create additional conditions. Ensure that you copy an entire “if” statement, up to and including the “<?end if?>”

Example

```
<?if:number(INV_AMOUNT)>10000?><?attribute@incontext:background-color;'#FFB9B9'?><?end if?>
```

Highlighting an Entire Table Row

To highlight an entire table row, follow the preceding instructions; however, ensure that you select **Apply to Entire Table Row**.

Search Parameters and Additional Parameters

When creating, you can specify two types of parameters before running a report:

- ▶ Search Parameters
Use these to filter the SQL results. Data View result rows that do not match the filter will not be sent from Unifier to BI Publisher.
- ▶ Additional Parameters
Use these to pass additional parameters for BI Publisher to consume at runtime. All values are sent from Unifier to BI Publisher for further operation.

Search Parameters

Search parameters allow the report runner to filter the data that gets sent to BI Publisher. For example, a Contract Report may need to be filtered by Vendor or Contract Type. A Ball-in-court report may be filtered for a specific task assignee, or task type.

At runtime, the report runner can select or multi-select from a list of values based on the search parameters specified in the report configuration. This means that Unifier basically runs the Data View before the report is run to give the report runner a list of the SQL results. He can then select or multi-select values from this initial run to filter what will appear in the report output. For more information on the Search Parameters, see "Add query parameters to custom report (Query tab)" in the Unifier Help.

Example

The following are instructions for filtering your Invoices Report on Invoice Vendor:

- 1) Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **Configuration** > **Custom Reports**.
- 2) Put your Invoices Custom Report in Draft mode and open the report.

- 3) Click on the **Query** tab.
- 4) Click **Add**.
- 5) Populate the field as follows:
 - ▶ Element Name: VENDOR
 - ▶ Label: Vendor
 - ▶ Operator: Equals
 - ▶ Source Type: View
 - ▶ Select View: Invoices
 - ▶ Value Column: VENDOR
 - ▶ Label Column: VENDOR
 - ▶ Selection Type: Multiple
- 6) Click **OK** to add the Query.
- 7) Click **OK** to save your changes.
- 8) Publish the report.

Note: You can only add queries on your Main View (not Sub Report Views).

Upon running the report, the report runner display the Invoice Query window. When you click Select, Unifier runs the Data View that you selected (Invoices) to get a list of all the VENDORS that would be returned if there were not filters. Multi-select your Vendors by using the Ctrl key. Enter text in your report to indicate that it was run using a query. We can access the value specified by the report runner via a top-level field.

Additional Parameters

Additional parameters allow the report runner to specify parameters that can be used at report runtime to alter how the data is displayed.

Unlike Search Parameters:

- ▶ Additional Parameters only support entering text. You cannot select or multi-select values from Unifier.
- ▶ Additional Parameters allow us to specify a default value so that the user only needs to change the value if needed.

A good example of where to use Additional Parameters is for calculations, for example, when you enter a target future exchange rate. You can also use Additional Parameters for conditional highlighting so that only rows over/under a certain value are highlighted.

Note: In the section Conditional Formatting, You need to hard-code a value (for example, 10,000).

The following are instructions for conditionally highlighting Invoices based on amount in our Invoices Report:

- 1) Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **Configuration** > **Custom Reports**.
- 2) Put your Invoices Custom Report in Draft mode and open the report.
- 3) Click the **Parameters** tab.
- 4) Click **Add**.
- 5) Populate the field as follows:
 - ▶ Row: 0
 - ▶ Editable: [Select]
 - ▶ Hidden: [Do not select]
 - ▶ Name: Amount
 - ▶ Label: Highlight Amount
 - ▶ Default: 10000

Note: When running the report, the selections add parameter called 'Amount', with label 'Highlight Amount' and an editable value of 10000 in the Additional Parameters box.

- 6) Click **OK** to save your changes.
- 7) Publish the report.

If you run the report at this stage, you see no changes because unlike Search Parameters, all the same data is passed to BI Publisher. You need to add some additional logic to your RTF template to use this new 'Amount' parameter.

- 1) Sign in to **Unifier** as a Company Administrator and go to the **Company** tab (Company Workspace), ensure that you are in Administration mode > **Configuration** > **Custom Reports**.
- 2) Put your Invoices Custom Report in Draft mode and open the report.
- 3) Download the new Sample Data for your project. A new top-level parameter is generated in your Sample XML: `<uuu_p_Amount>10000</uuu_p_Amount>`
- 4) Open your RTF template and import the new Sample XML file so you can add the logic.
- 5) Define a new variable somewhere at the top (as we did in Defining the Variables) as follows:
`<?variable@begin:uuu_p_Amount;(./uuu_p_Amount)[1]?>`

To use the variable, you can call `$uuu_p_Amount` instead of your hard-coded 10000 in our conditional statement (assuming you already followed the instructions in Conditional Formatting).

- 1) Double-click on the 'C' Conditional Formatting field that you added to the Amount cell in the Conditional Formatting section
- 2) Ensure that you see the following on the Advanced tab:
`<?if@row:number(INV_AMOUNT)>10000?><?attribute@incontext:background-color;'#F8DADB'?><?end if?>`
- 3) Replace the '10000' with '\$uuu_p_Amount':
`<?if@row:number(INV_AMOUNT)>$uuu_p_Amount?><?attribute@incontext:background-color;'#F8DADB'?><?end if?>`
- 4) Click **OK**.

If you run the report, you see the same behavior as before because the value in your Sample XML is the same as the value you hard-coded before (10000). To test:

- 1) Open the Sample XML file that you have downloaded (in notepad) and update the value of `uuu_p_Amount` to a value lower (for example, 5000).
- 2) Save the XML file and re-run the report. The Invoices over \$5,000 are now highlighted in red.
- 3) Upload the new RTF template into your Custom Report in Unifier.
- 4) Click **OK** to save your changes.
- 5) Publish the report.
- 6) Proceed to test the Custom Report in Unifier. Since the 'Amount' parameter is editable, you can update the parameter before running the report.

Adding Permissions to Run BI Publisher Reports in Primavera Unifier

These steps apply to sample custom reports and may not apply for every report.

To add permissions to run reports:

- 1) Go to the **Company Workspace** tab and switch to **Admin** mode.
- 2) Select the **User Mode Navigator**.
- 3) Open the **Project / Shell** navigator.
- 4) Under **Reports**, create a **Custom Reports** node.
- 5) Change the icon to `u_report_select.gif`.
- 6) Under the **Custom Reports** node, add the new report.
- 7) Click **OK**.
- 8) Highlight the **Project / Shell** navigator.
- 9) Select **Deploy** to apply changes.
- 10) Go to the **Company Workspace** tab and switch to **Admin** mode.
- 11) Navigate to **Company Sponsored Shells, All Projects**.
- 12) Highlight **All Projects** in the right window pane and select **Open**.
- 13) Select **Access Control**.
- 14) Navigate to **User Mode Access, Reports, Custom Reports**.
- 15) Select the custom report.
- 16) Add users to allow access to reports.
- 17) Click **Apply** and **OK**.

Running a Report in Primavera Unifier

To run the BI Publisher Report in Primavera Unifier:

- 1) Sign in to your Unifier environment.
- 2) Select a Project or Shell.
- 3) Ensure you are in **User** mode.
- 4) Under the Project/shell navigation tree, select **Reports, Custom node**.

- 5) Select a report and a dialog box will open.
- 6) In the dialog box, enter the search conditions and click **Report** to run the report.

Legal Notices

Oracle Primavera Cloud Services Reporting Administration Guide

Copyright © 2016, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Xeon are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Opteron, the AMD logo, and the AMD Opteron logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, delivered to U.S. Government end users are "commercial computer software" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, use, duplication, disclosure, modification, and adaptation of the programs, including any operating system, integrated software, any programs installed on the hardware, and/or documentation, shall be subject to license terms and license restrictions applicable to the programs. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate failsafe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

This software or hardware and documentation may provide access to or information on content, products and services from third-parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services.