PI RtReports Administration Online

OSIsoft, LLC
1600 Alvarado Street
San Leandro, CA 94577

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, photocopying, recording, or otherwise, without the prior written permission of OSIsoft, LLC.

OSIsoft, the OSIsoft logo and logotype, Managed PI, OSIsoft Advanced Services, OSIsoft Cloud Services, OSIsoft Connected Services, OSIsoft EDS, PI ACE, PI Advanced Computing Engine, PI AF SDK, PI API, PI Asset Framework, PI Audit Viewer, PI Builder, PI Cloud Connect, PI Connectors, PI Data Archive, PI DataLink, PI DataLink Server, PI Developers Club, PI Integrator for Business Analytics, PI Interfaces, PI JDBC Driver, PI Manual Logger, PI Notifications, PI ODBC Driver, PI OLEDB Enterprise, PI OLEDB Provider, PI OPC DA Server, PI OPC HDA Server, PI ProcessBook, PI SDK, PI Server, PI Square, PI System, PI System Access, PI Vision, PI Visualization Suite, PI Web API, PI WebParts, PI Web Services, RLINK and RtReports are all trademarks of OSIsoft, LLC.

All other trademarks or trade names used herein are the property of their respective owners.

U.S. GOVERNMENT RIGHTS

Use, duplication or disclosure by the US Government is subject to restrictions set forth in the OSIsoft, LLC license agreement and/or as provided in DFARS 227.7202, DFARS 252.227-7013, FAR 12-212, FAR 52.227-19, or their successors, as applicable.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, mechanical, photocopying, recording or otherwise, without the written permission of OSIsoft, LLC.

Software Versions Used in this Document

The list below describes the software versions used in this version of the course.

Software	Version
PI Data Archive	2018
PI Asset Framework	2018
PI RtReports	4.1.0

Contents

Lesson 1: PI RtReports Overview 6
PI Reports Components 6
Why Use RtReports? 7
Architecture & Workflow 9
RtReportsContexts 11

Lesson 2: Building Batch Reports 12
Batch Based Templates 12
What Is a Unit Class? 15
Exercise – Configure Unit Classes 16
Solution: Configure Unit Classes 16
The Batch Journal 17
The Batch Template Wizard 18
Creating Batch Report Templates 19
Solution: Configure Batch Report Templates 20

Lesson 3: Data Formatting 23
The Default Template 23
Formatting Elements 23
Hierarchies 24
Formatting 24
Exercise – Batch Report Format Template Editing 26
Solution: Batch Report Format Template Editing 28

Lesson 4: Using Iterations 33
Working with Iterator Expressions 33
Where is the Iterator Expression? 34
Multi-Level Iterators 34
Exercise 4: Adding Iterators to the Format template 36
Exercise 4 Solution: Adding Iterators to the Format Template 38

Lesson 5: Journal Actions 39
Context 39
Action Parameters and Results 39
Exercise 5 – Adding a Journal Action 41
Exercise 5 Solution: Adding a Journal Action 43
Exercise 6 – Add Compressed Data 48
Exercise 6 Solution: Add Compressed Data 50

Lesson 6: Time Based Reporting 55
Time Based Templates 55
The Time Journal 56
Using Equipment in a Time Report 57
Exercise 7 – Time Report Templates 58
Exercise 7 Solution: Time Report Template Creation 60
Lesson 7: Condition Based Reporting 66
Using Action States 67
1.1 Exercise – Using the State Machine 70
Exercise 8 Solution: Using the State Machine 71
Using Criteria 77
Exercise 9 – Using Conditional Formatting 80
Exercise 9 Solution: Using Conditional Formatting 82
Lesson 8: Finishing Touches 85
Toolboxes 85
Exercise 10 – Using Toolboxes: Predefined Actions and Style Sheets 87
Exercise 10 Solution: Predefined Actions and Style Sheets 89
Defining Printing 92
Versioning 92
Lesson 9: The PI RtReports Generator 96
Import Export Reports 96
Exercise 11 – Import a Batch-Based Report 97
Exercise 11 Solution: Import a Batch-Based Report 98
Exercise 12 – Import a Time-Based Report 104
Exercise 12 Solution: Import a Time-Based Report 105
The RtReports Generator 108
Generating Batch Reports 108
Configure automatic report generation 114
Exercise 13 – Generating a Batch Based Report 115
Exercise 13 Solution: Generating a Batch-Based Report 116
Exercise 14 – Generating a Time-Based Report 121
Exercise 14 Solution: Generating Multiple Time-Based Reports 122
Run Time Action Parameters 125
Implementing Run Time Parameters 125

Exercise 15 – Run Time Parameters for Batch Report 126
Exercise 15 Solution: Run Time Parameters for Batch Report 127

Lesson 10: PI RtReports Security 130
Introduction 130
RtReports Windows Groups 130
Template Security 131
Report Template Review Cycle 132
Configure an Approval Plan for a Generated Report 133
Exercise 16 – Group and User Security 134
Exercise 16 Solution: Setup Group and Users Security 135

Lesson 11: Miscellaneous Topics 143
PI Batch Interfaces 143
Template Placeholders 143
PI Baseline Service 145
Introduction 145
Data Sources 145
Server side Configuration Files 147

Batch Reference Information 150

PI Batch Database Architecture 151
Introduction 151
What is the PI Batch Database? 151
PIBatch 152
PIUnitBatch 152
PISubBatch 153
How PIBatch/EF Maps to the S88 Standard 153
Examples of Batch 154
Exercise – BatchView 155

PI Module Database 156
Modules 156
Heading Sets 156
Aliases 157
Properties 157
PI Module Database Structures Used in PI Batch 157
PI Module Database Tools 158
Exercise – Configure *PI Modules* for Pulp and Paper Line 1 159
Directed Exercise – Configure PI Batch for Pulp and Paper Line 1 160
Exercise – Configure PI Batches for Pulp and Paper Line 2 161

Solution: Configure PI Batch for Pulp and Paper Line 2 162
Exercise – Configure PI Batch for Cake Making 164
Solution: Configure PI Batches for Cake Making 166

PI AF, Asset Analysis and Event Frames (EF) 167
What does Asset Framework (AF) do for me? 167
Key Features of the AF Server 169
PI System Explorer 170
Activity – Search Event Frames (EF) 174
Exercise – Configure AF Elements (Assets) for Pulp and Paper Line 1 176
Exercise – Configure EF for Pulp and Paper Line 1 177
Exercise – Configure EF for Pulp and Paper Line 2 179
Solution: Configure EF for Pulp and Paper Line 2 181
Exercise – Configure EF for Cake Making 183
Solution: Configure EF for Cake Making 185

Lesson 1: PI RtReports Overview

RtReports is a ground-breaking process reporting solution that enables the production reports based on both real-time and historical data from internal PI system data source and external relational data sources gathered by RtBaseline Services. Report templates may be based on Batch (Event Frame or PI Batch) or time range.

For batch process manufacturers, RtReports is a comprehensive system to generate electronic batch reports that comply with regulatory requirements – incorporating all steps from data collection through report generation to sign-off.

For continuous manufacturers, RtReports offers reports based on hierarchical time frames or interrelated time ranges.

RtReports maintains all data at its original fidelity, offers a full audit trail of data, user security changes, features stringent versioning to remove errors, align reports with production, and support auditing. RtReports supports electronic signatures for efficient report approval. Authorized users can access reports online by means of secure Web services.

PI Reports Components

RtReports is not just one single entity. It is made up of a series of components that all work in harmony.

RtReports Server

Using IIS and ASP.NET, the RtReports Server is serving the data to the RtReports Generator request. The PI Baseline service allows users to access data from a Relational Database server.

RtReports Editor

The Editor is a .Net windows application, it allows users to create and modify report specifications (called Report Templates) for selecting data, aggregating it, formatting it, and collecting electronic signatures for reviews.

RtReports Generator

The RtReports Generator (using Web Browser) is a .Net web application, it allows users to search data within a PI AF Server and/or PI Data Archive server, aggregate that data according to specification, identify deviations from critical process limits, maintain comments and approval history, and generate pre-configured reports.

Why Use RtReports?

FDA Regulation 21 CFR Part 11

The pharmaceutical and biotechnology industries have been investing in paperless systems since the 1980's, and for compelling economic reasons. These systems are used to automate, monitor and report on production operations, and are an integral component of Good Manufacturing Practice.

As such they are subject to FDA regulation. During the 1990's, the technologies available for electronic record processing and authentication reached the point where the industry needed specific guidelines for the FDA's acceptance of paperless systems, and the FDA responded in March 1997 with the regulation known as 21 CFR Part 11.

21 CFR Part 11 is concerned with the authenticity, integrity and confidentiality of electronic records and signatures. Computer systems must be validated to ensure the accuracy, reliability, and consistency of operations, and their ability to identify invalid or altered records. A key requirement for compliance is the ability of the system to create an audit trail, or audit management database. This database must, provide a secure log of all attempts to access, or log onto the system, together with the operations performed.

- The audit management database must enforce these criteria:
- Audit records are secure, computer generated and time-stamped
- The person making any change must be identified
- Both original and changed data must be recorded
- Evidence of previously changed or deleted data must be visible
- FDA must have access for review purposes

21 CFR Part 11 is highly significant because it allows electronic records to be considered equivalent to paper records, and electronic signatures to be considered equivalent to handwritten signatures. The regulation is subject to phased implementation with the initial focus on drug manufacturing. The FDA is expected to stop accepting paper records in the near future, and while there has been a temporary grace period, FDA inspectors are now trained in the regulation, and enforcement has begun. Existing computer systems must be updated or replaced in order to comply.

Simplifying Title V Compliance

Title V of the Clean Air Act creates a new world of regulatory environmental compliance. However, this watershed event is neither universally recognized nor appropriately funded by Management. Title V operating permits require facility

owners to certify environmental compliance. That means the burden of compliance monitoring is no longer on the regulatory agencies—it is now on the owner/operator.

To meet Title V requirements, companies must now keep records that prove compliance, plus issue exception reports. Plants and entire enterprises saddled with diverse information systems and complex manufacturing conditions are seeking enterprise-wide data visualization and reporting solutions to meet Title V regulations.

Harnessing the right information at the right time can make or break your ability to comply with Title V requirements. OSIsoft's infrastructure based on the PI System™ as its core engine, delivers unified information across the enterprise to multiple functional disciplines, from management to environmental and operations. This information is vital to meeting regulatory requirements while simultaneously reducing compliance costs.

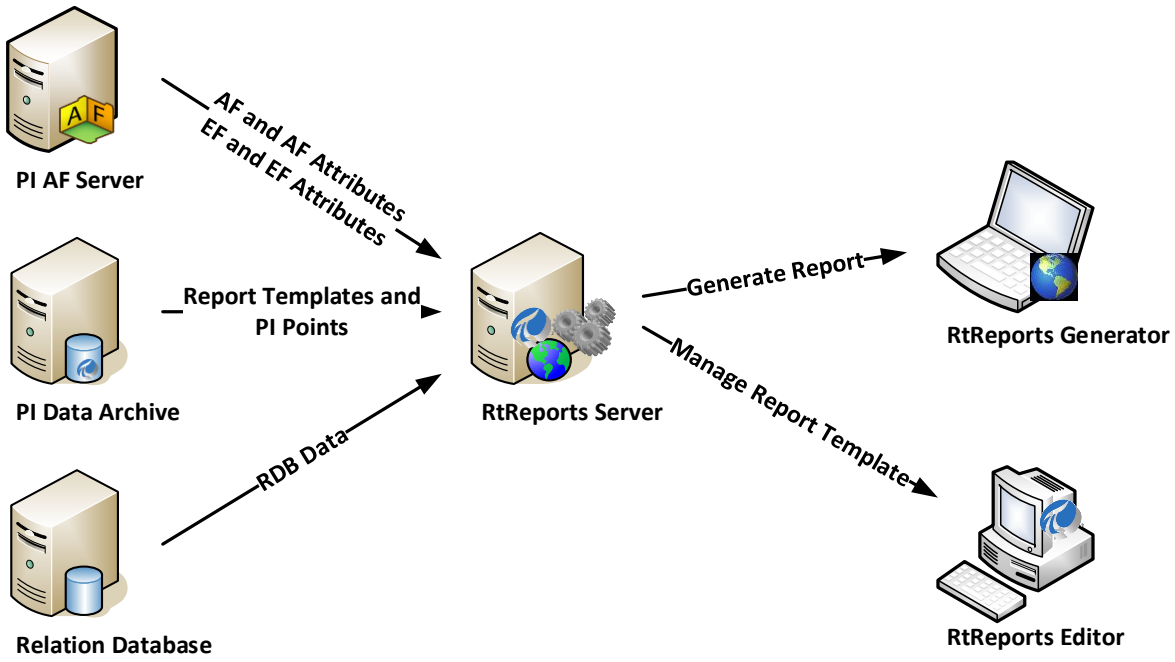
This Webinar presents typical compliance data required by Title V and shows how OSIsoft's PI Infrastructure captures this data and presents it enterprise-wide. The Webinar also highlights an implementation case study at Wasatch Energy Systems.

Wasatch Energy Systems Case Study
Wasatch Energy Systems of Layton, Utah, uses OSIsoft's PI System to meet regulatory requirements for monitoring, recording, and reporting environmental data related to NOx, SOx, CO, and other emissions. This presentation describes plant and Continuous Emission Monitoring equipment and the PI System, as well as exploring regulatory requirements and showing how PI enables compliance. You'll learn how PI addresses system availability, data reporting, and auditing along with details of project execution.

Architecture & Workflow

The Architecture provides for the following:

- Auditing
- Formatted Reporting
- Secure Reporting (pdf)

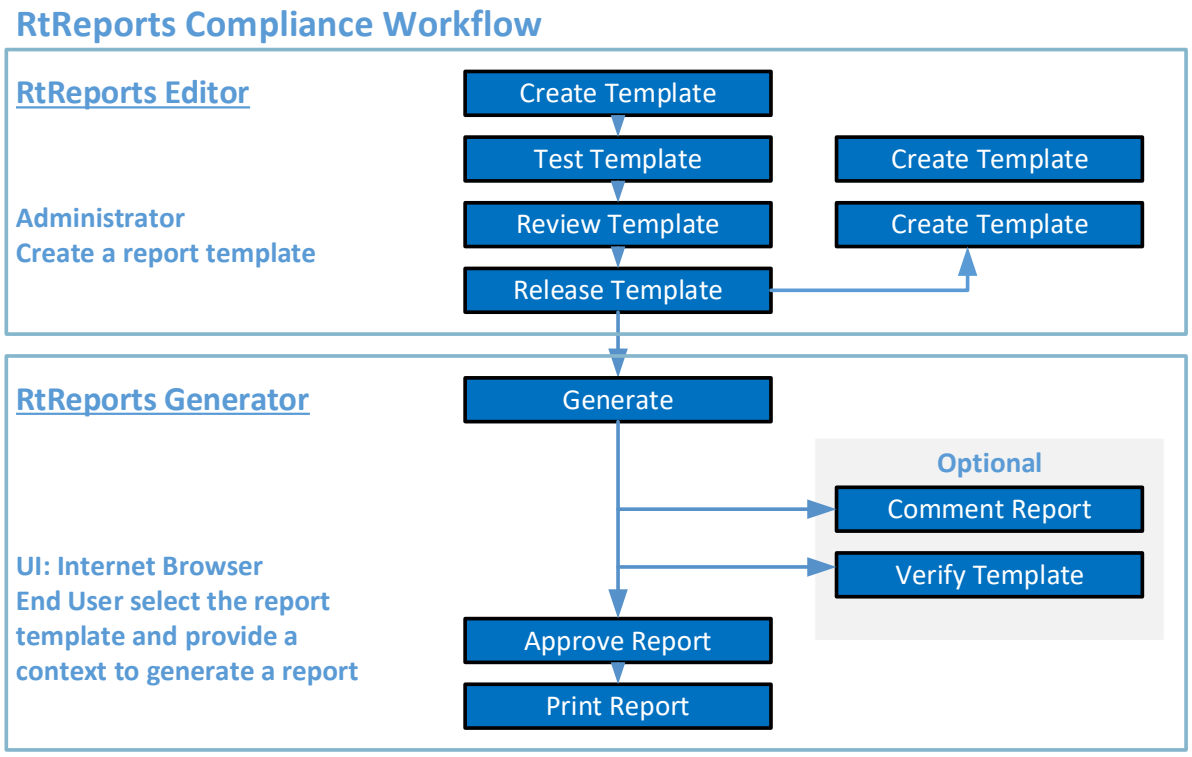


Preferred Architecture – Better Performance and Security

Workflow

The RtReports workflow has many benefits, including:

- Report Template Status – editing, released
- Security and Approval routes for Editing
- Security for Report Template Generation
- Security and Approval routes for commenting, printing, verifying sections, and final report approval
- All inputs require electronic signature



Report Template Status

When a Report Template is created, it is created with a status of “In Edit”. In order for general users to execute a Report Template, it must have a status of “Released” and they must have been given explicit Execute permission. You change the Report Template status from “In Edit” to “Released” by successfully completing a review cycle.

Report Template Review Cycle

A review cycle controls changing the Report Template status to “Released”. At each step in the Review Cycle, comments and electronic signatures are gathered and stored with that version of the Report Template. At any stage in the Review Cycle, a user can reject the Report Template which will return the status back to “In Edit”.

Report Comments

Since the executed report is meant to serve as your permanent manufacturing record, you may need to add in-line comments to explain certain entries within the report. You can use the comment feature within RtReports to associate comments to each section of the executed Report.

Report Verifications

To conduct real time review of your executed report, you can use the verification feature within RtReports to associate one or more electronic signatures to each section of the executed report. This can act as your first reviewer of your manufacturing record.

Report Approval Plan

RtReports tracks the status of Report Template and its associated context. For example, every time you run a weekly report for an official time range, that association and status is in the configured PI Server.

If a Report Template has never been run against a specific context, it will have a report status of “Not Run”. If a Report Template has been run against a specific context, it will have a report status of “Not Yet Approved”. The Approval Plan for a Report Template controls the process of changing the report status from “Not Yet Approved” to “Approved”. At each step in the Approval Plan, comments and electronic signatures will be gathered and stored with that specific report and context. When the Report Approval Cycle is complete, the report status will be “Approved”.

RtReportsContexts

Batch/EF Template

Batch Templates (same for EF Templates) allow users to configure Report Templates that can be generated for a specific Procedure (i.e. PIBatch or EF) from the PI Batch or EF Database. Each Procedure includes all the sub-items such as the Unit-Procedures, Operations, and Phases.

Time Report Templates

Time Report Templates allow users to configure Report Templates that can be generated for a specific time range or recurrence pattern. The Report Template is generated against an overall time range but can be further divided into relative or fixed duration time spans. Users can now create detailed time reports with hierarchical time frames or interrelated time ranges.

Lesson 2: Building Batch Reports

PI Batch Templates allow users to configure Report Templates that can be generated for a specific PIBatch from the PI Batch Database. It also includes all the sub-items such as the PIUnitBatches and PISubBatches. Most likely, the batch database will be populated by the Event File or the PI Batch Generator Interfaces.

Note: You will be able to do everything without being prompted because you are logged in as an administrator and the security has been configured to allow unrestricted access. This is to make the classroom exercises easier – ***however this is not the default behavior***. Security will be addressed later in the class.

Batch Based Templates

Report Template criteria define the relationship between the PI Batch Database and the Report Template. Use the criteria settings below to associate a Report Template to a recipe, product, equipment, SubBatch, Unit Class, or a combination of criteria.

The second step of building an RtReports template consist in defining the business rules that will be used to retrieve data from PI Servers, Relational Data Sources and Web Services. This task is conducted under the Data Template Pane.

For a Batch Template Report, the goal of the Data Template is to define which time intervals are going to be used along with the data that should be presented into the report. The time intervals are generally going to be expressed by PIBatch, PIUnitBatches and PISubBatches.

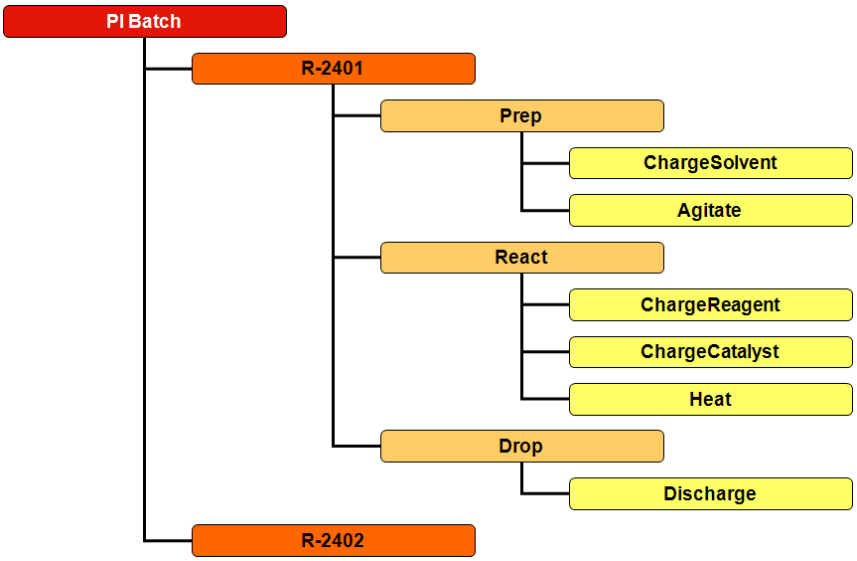
Defining Data Retrieval through the Data Template

The Data Template stores business rules. These rules control how RtReports retrieves data from the PI Archive. RtReports Generator is responsible for interpreting the business rules defined for the Report Template, executing them, and creating a result set, which appears in the Journal tab.

Data may be gathered from the PI Batch Database, the PI Archive, the PI Module Database, a Relational Database or a Web Service. Next, the rules are processed and the “Journal” (unformatted results set) is created.

About Contexts

A Data Template has a context, or basic structure. A context is a set of key events that frame your report. This structure will closely resemble your process model or hierarchy. You create Batch Templates, Unit Templates (linked to a unit of equipment in the PI Module Database), and SubBatch Templates (operations and phases) to describe the mapping rules to include PI Batch Database objects in the report.

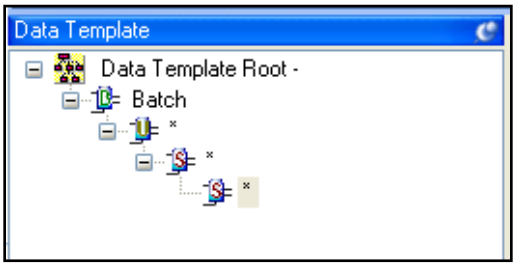


For example, suppose you want to create a report for a product produced by a batch run through Units R-401, R-402, and R-403. The major events within this report occur when the batch starts and ends in these units, as well as the start time and end time of the overall batch run. This “event framing” creates records within the PI Batch Database.

RtReports uses context elements to map important events wanted on the report to those corresponding events within the PI Server.

Extracting the Batch Hierarchy

Using a batch template, the report will always start at the PIBatch level. At this time, the user should decide which elements from the PI Batch Database he wants to see. It is possible to add a Unit Template to extract data from the PIUnitBatches under the PIBatch level. In a similar way, it is also possible to add a SubBatch Template to extract data from the PISubBatches included in the PIUnitBatches or another PISubBatch if we add additional SubBatch Template level.

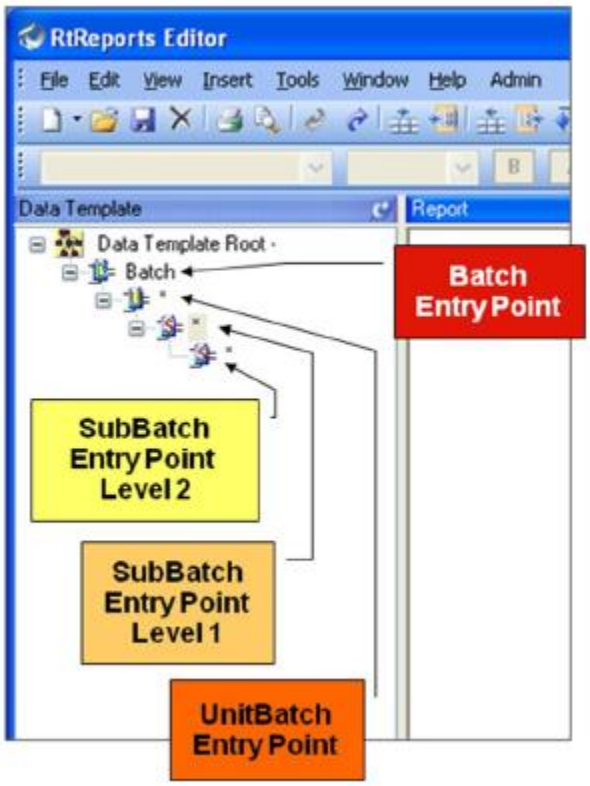


In every case, it is possible to use generic extracting business rules that will apply to all PIUnitBatches or PISubBatches. It is also possible to restrict the business rules to one or only a few PIUnitBatches or PISubBatches.

Applying the Business Rules

Applying the Business Rules to a context for a batch-based report means to run the rules against a specific PI Batch. This action will have as a result the creation of a journal.

In RtReports, the journal is the unformatted retrieved data. This information will be parsed into the Format Template for the final generation of the report.



All the data included in the journal is organized under an XML structure.

Note: When the Format Template section is configured, the Report pane will show a preview of the report.

To run the business rules against the context in a test use the test function, View -> Test Report -> Use PIBatch (F2) or Use Event Frames (F3). Use the Refresh (F5) function to view the effects of changes against the same data.

What Is a Unit Class?

Unit Classes allow you to group like equipment together and then reference the Unit Class definition. Using Unit Classes in conjunction with unit relative

Aliases allow you to reduce the number of Report Templates needed, which ultimately reduces testing and maintenance activities.

Creating a Unit Class is as simple as giving a name to the new class and adding individual units in a dialog box.

The Unit Class is applied at the Unit Template. If a Unit Template is class-based, the Generator will resolve a collection of all PIUnitBatches that belongs to the class defined. ***It then ignores the name pattern***.

Exercise 1 – Configure Unit Classes

Problem Description

Configure Chip Unit Class which consists of two units Chip1 and Chip2

Configure Digest Unit Class which consist of two units Digest1 and Digest2

Solution 1: Configure Unit Classes

Create a new Batch Report Template

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment using the Start > PI System > RtReports > RtReports Editor -> and login
2. Tools -> Unit Class Editor…
3. Click the green + above the Unit Class pane -> Unit Class Name: Chip -> Unit Class Version: 1.0 -> OK
4. Click the green + above the Unit Class pane -> Unit Class Name: Digest -> Unit Class Version: 1.0 -> OK
5. Select the “Chip” in the Unit Class pane -> click the green + above Units pane -> call it “Chip1” -> repeat clicking the green + above Units pane -> call it “Chip2”
6. Select the “Digest” in the Unit Class pane -> click the green + above Units pane -> call it “Digest1” -> repeat clicking the green + above Units pane -> call it “Digest2”
7. OK

The Batch Journal

The Journal holds the results of the Business Rules applied to a context, in this case, a PIBatch. This information is stored under an XML structure and it is important to understand its organization in order to extract and display the critical data in our reports.

At the top level, we have the BatchJournal. This level holds all the PIBatch details such as the Batch ID, Product, Recipe, Start and End Times, Duration. You can also see the Display Name and Description field of the report.

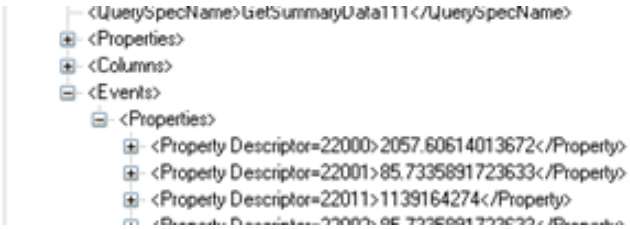
The PIUnitBatch data is located under the UnitBatchJournals which has a collection of UnitBatchJournal. Each one of these UnitBatchJournal holds the information of one specific PIUnitBatch such as Batch ID, Product, Procedure Name, Start and End Times, Duration and the Unit Name.

The PISubBatch data is located under the SubBatchJournals which has a collection of SubBatchJournal. Each one of these SubBatchJournal holds the information of one specific PISubBatch such as the Name, Start and End Times.

A SubBatchJournal can also hold a SubBatchJournals subdivided into SubBatchJournal for a second and the following SubBatch level.

Properties, Columns, and Events

Results are often returned as combinations of Properties, Columns, and Events.



These are all related and make up the whole of the result. For example, the Properties may contain a line item for the Maximum calculation result, the Columns will be a reference to the Descriptor number for the Maximum, and the Events will contain the actual value listed by Descriptor number. So by combining all three pieces we get the label “Maximum” and the actual result value for that label.

Walking Up the Journal

There are times when users need a piece of information in a report that is “higher” in the Journal Tree than the level users are using. For example, users are in an iteration at the *Operation* level and users want to cite the BatchID at the *Procedure* level. You will use the following syntax to walk “back” up the tree:

```
../../../../BatchID
```

This would retrieve the BatchID from two levels up in the hierarchy.

Terminal States vs. Sequence Entries

Nothing as far as the data is concerned. *Terminal States* results are grouped under the Action State Name and *Sequence Entries* results are listed in chronological order of the Action State. They contain the same data.

So, which one should users use? The answer is simple, if users are iterating and the order events (states) matter then use Sequence Entries; if users are iterating and grouping Results under event (states) then use Terminal States.

The Batch Template Wizard

You may want to build up your Data Template structure using a batch that has been previously processed. You can accomplish this by using the Data Template Builder. The Data Template Builder provides searchable access to the PI Batch Database. You can search for batches that meet your specified criteria. Once you have found a batch, you select that batch and then you may select or deselect the processing steps that you want included in the report.

Exercise 2: Creating Batch Report Templates

Objectives

- Learn how to create a new Batch Report Template using the *RtReports Editor*.
- Learn how to add Unit, PISubBatch, and PISubBatch of PISubBatch Templates.

Problem Description

You are asked to create a report template for a batch process. This exercise will focus on the construction of the Data Template section. Your report should be built in a way that you can extract data from the PI Batch, PI UnitBatch and PI SubBatches level.

Approach

Create a new *Batch Report Template* called Batch_Report.

Add a *Unit Template -> Name: *.*

Add a *SubBatch Template -> Name: *.*

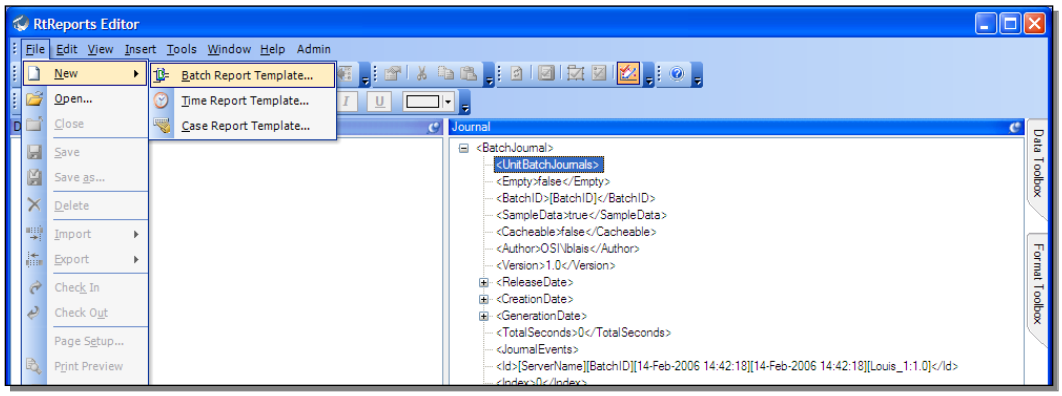
Add a *SubBatch Template of a SubBatch.-> Name: *.*

Test Report using EF (F3)

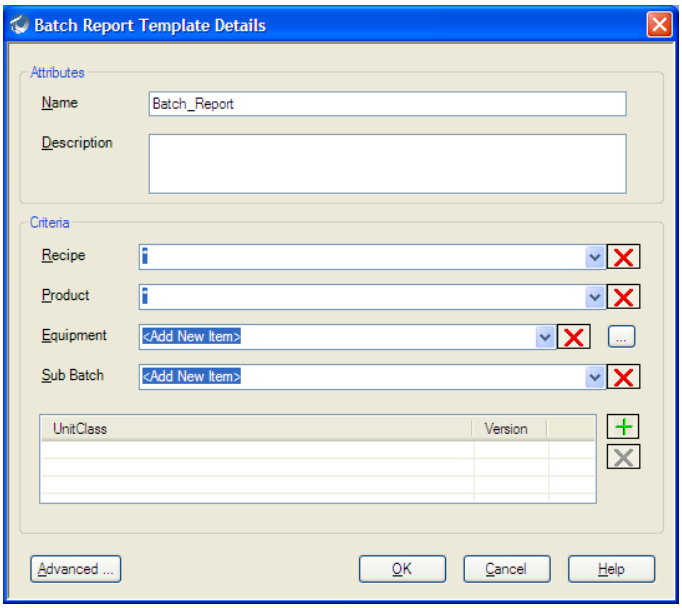
Solution 2: Configure Batch Report Templates

Create a new Batch Report Template

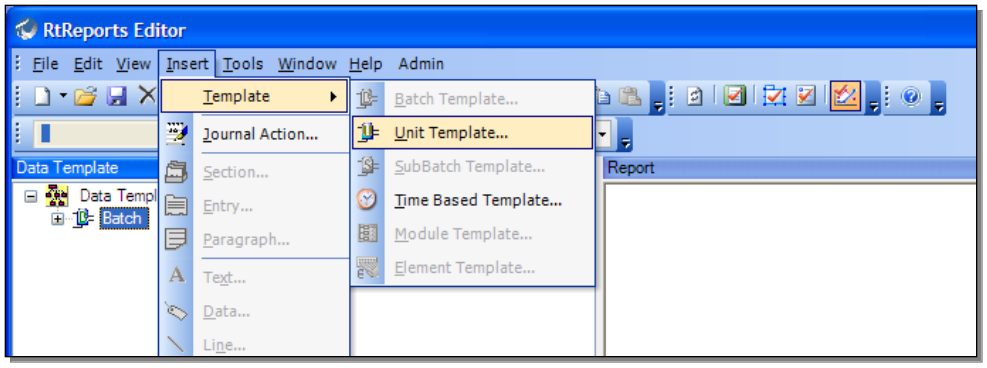
1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment using the Start > PI System > RtReports > RtReports Editor -> and login
2. Select the *Data Template* tab
3. There are two ways to do so:
4. Select File ⟶ New ⟶ Batch Report Template…
5. Select the New Icon ⟶ Batch Report Template…



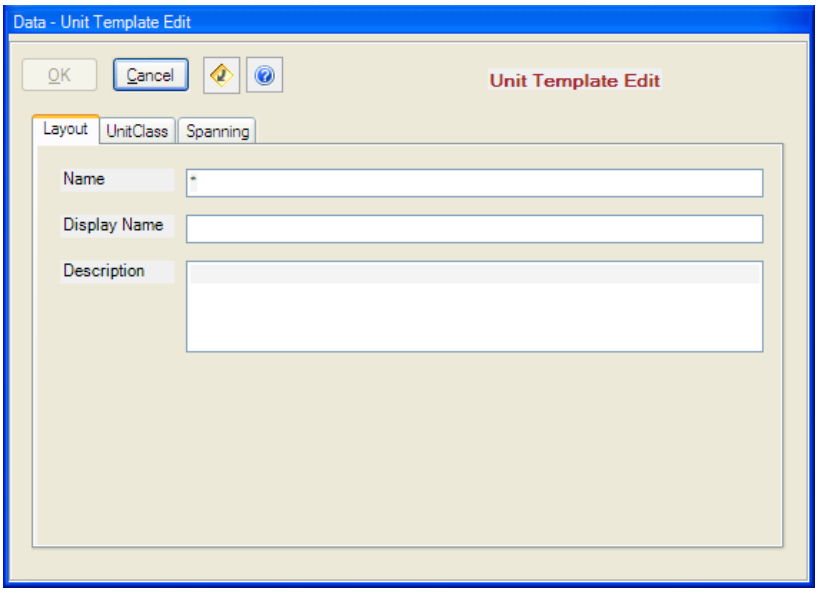
6. In the name field, type in: Batch_Report



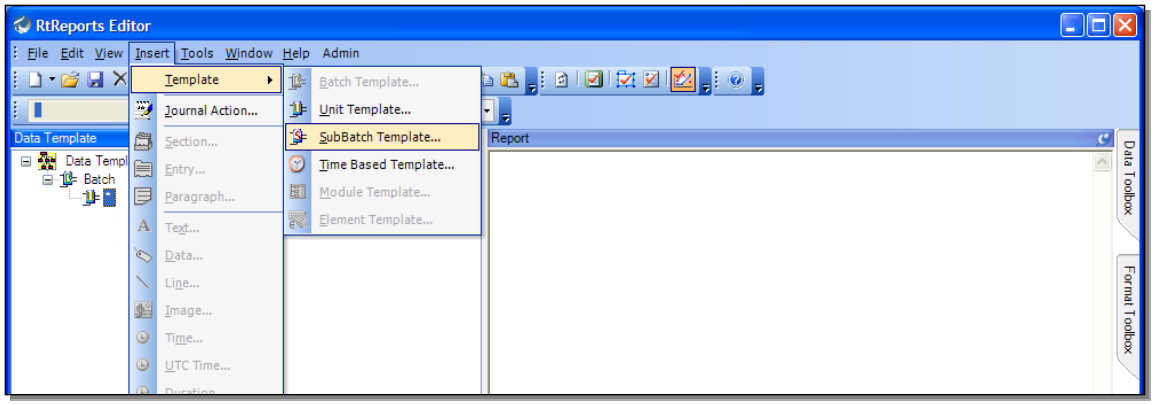
7. Leave the remaining boxes unchanged -> *OK*
8. Add a Unit Template. There are 3 ways to do so:
   - **a.** Right-click on *Batch* -> select Insert and click on *Unit Template*.
   - **b.** Select *Batch* by clicking on it, then click on *Insert*, select *Template* and click on *Unit Template.*
   - **c.** Hover the mouse pointer over *Data Toolbox* on the right-hand side of the screen, click and drag *Unit Template* and release it over *Batch*



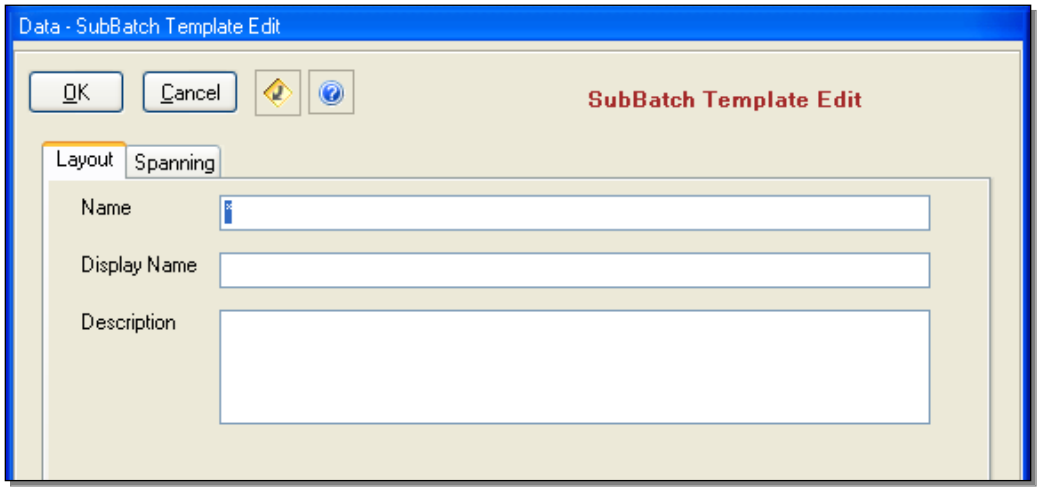
9. Make sure that the *Name*: *



10. Leave the other boxes empty -> *OK*
11. Add a SubBatch Template. There are 3 ways to do so:
    - **a.** Right-click on *, select Insert and click on *SubBatch Template*.
    - **b.** Select * by clicking on it, then click on *Insert*, select *Template* and click on *SubBatch Template*
    - **c.** Hover the mouse pointer over *Data Toolbox* on the right-hand side of the screen, click and drag *SubBatch Template* and release it over *



12. Make sure that the *Name* section states *



13. Leave the other boxes unchanged -> *OK*

Save your work using the File -> Save menu

Lesson 3: Data Formatting

Once your Data Template is built, you have all the Business Rules required to extract data. These rules are applied to a context and the outcome of this operation is the Journal. The data to report on is then gathered but not presented. In the Format Template, we will see how we can take the information from the Journal and present it in the report. At this point, we can decide on which information should go in the report and how we want to display it.

This will mainly be done by adding formatting object to the report and by using the elements of the Journal as the source of these template objects.

The Default Template

The main organizational element is the Section element. By default, when a Report Template is created, it contains 5 specialized sections: Cover Sheet, Report Header, Page Header, Page Footer, and Report Footer. You build up the report content structure by adding additional sections to the Report Template that breaks your report into logical groups.

You further divide a Section element by adding Entry elements or additional Section elements. A Section element contains several additional components that an Entry element does not, such as a Section Header and Section Footer. The 5 specialized sections named above cannot contain additional Section elements. They can only contain Entry elements.

After you have an Entry element in the Reports Content tree, you can add Drawing elements to the Entry. Examples of drawing elements are Text, Image, Table, Line, etc.

Formatting Elements

There are 17 formatting elements that can be divided into three main categories, Grouping, Static Drawing, and Dynamic Drawing elements.

Grouping Elements that gather other elements

Section: separates the report into logical groups. Comments and verification tasks can be associated to sections attached to the Root Node.

Entry: separates sections into logical groups.

Static Drawing **Element that are set at design time**

- Text: draws static text on the report.
- Line: draws a horizontal line on the report.
- Image: object that provides the inclusion of a graphic within the report.
- Page Break: object that provides page breaks within the printed report. Not visible through Generator.
- Page Num: objects that provide page position within the printed report. Not visible through Generator.
- Table, Table Rows, and Table Data: objects that allow you to build tables within the report.

Dynamic Drawing Elements that are linked to an elements of the Journal

- Data: adds numeric and string data from the Journal to the report.
- Time: adds a Time object from the Journal to the report.
- Paragraph: provides formatting within an Entry.
- UTC Time: adds a UTC Time string from the Journal to the report.
- Duration: adds a duration string from the Journal to the report.
- Trend: adds a Trend to the report.
- Trace Set: adds a table directly from data within the Journal to the report.

Hierarchies

Normally you will be held to a certain hierarchy. That is, you will first add a Section, and then and Entry, and then an Element, and then sub-elements to display your data. For example, you want to include the Batch ID in your report. You would first add or use an existing section, then add an entry like a paragraph, and then you would add the data element.

Specific formatting can be applied at any level.

Formatting

Almost everything that is visible can be controlled through some sort of formatting setting. The followings are some of the items that have controllable definitions:

- Text alignments, fonts, colors, size
- Numbers (shown below)
- Time and Date (shown below)
- Paragraph Formats
- Page breaks
- Lines
- General Report Info (name, version, generation time, etc.)
- Report Template Root Properties (include margins, headers, page orientation)

Number Formats	**Name**	**Example**
C or c	Currency	C2 => $76.43
E or e	Scientific	E3 => 7.643E+001
F or f	Fixed-point	F4 => 76.4315
P or p	Percent	P2 => 7,643.51%

Date and Time Format	**Name**
yy, yyyy	Year, 2 or 4 digits
M, MM	Month, Short/Long number
MMM, MMMM	Month, Short/Long name
d, dd	Day, short/long number
ddd, dddd	Day, short/long name
h, hh – H,HH	Hour, short/long number (12h or 24h clock)
m, mm	Minute, short/long number
s, ss	Second, short/long number
z, zz	Time Zone, short/long name

Exercise 3: Batch Report Format Template Editing

Objectives

Discover RtReports Report Formatting objects, such as Section, Entry, Data, UCT Time, Table…

Discover RtReports Report Preview pane

Problem Description

You must create a report for a daily batch. You have already built the *Data Template*. You must now edit the *Format template* to make the report display your results. You first need to create a cover page and then to add a table in the body of your report.

Approach

Open the Batch_Report from previous exercises

Select the *Format Template* tab.

Add a *"Main Section" Section* between the header and the footer.

Insert a *"CoverContent"* Entry into the *CoverSheet* section.

Insert a text object in *CoverContent* that states *OSIsoft Test Report,* using Arial font, bold formatting and 24 pt. police.

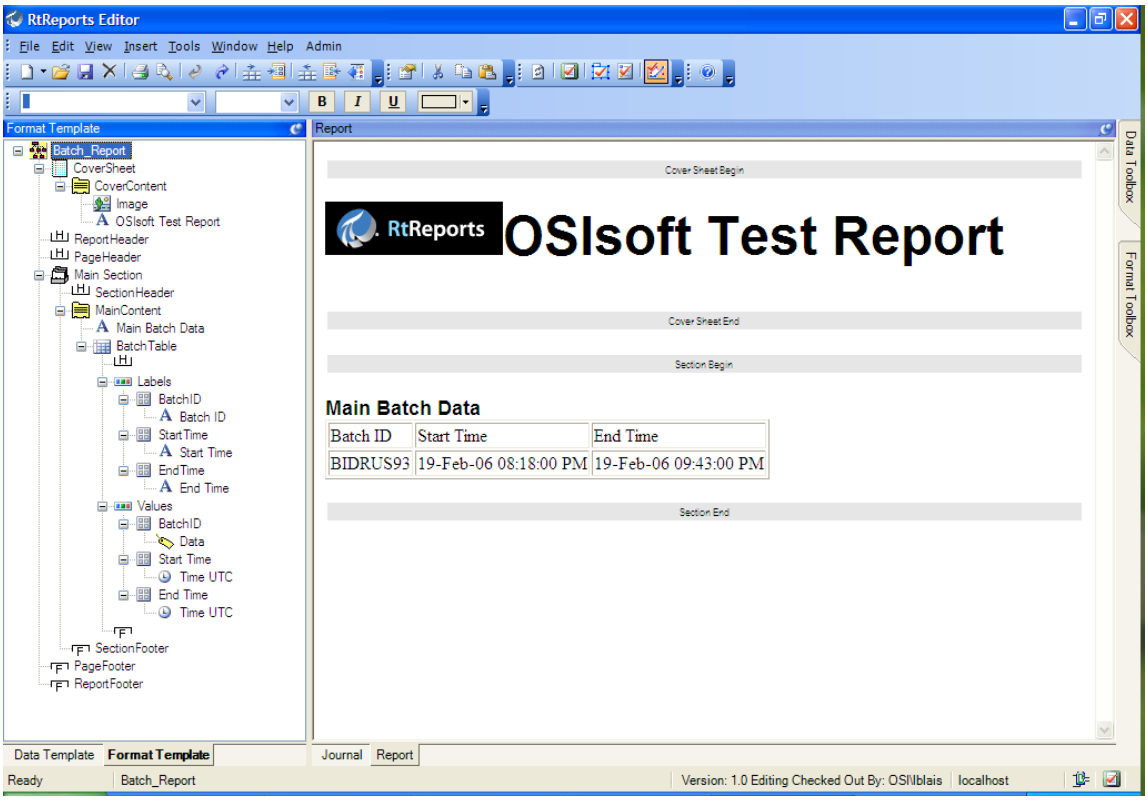
Insert the image RtReports_01.gif from *%PIHOME%\RtRetports\images* in *CoverContent*.

Insert a table that looks like the following under the entry of the *Main Section*.

Test the report using EF of Batch ID stem "BID"

BatchID	Start Time	End Time
BIDBDM28	15-Feb-06 06:43:40 AM	15-Feb-06 07:50:03 AM

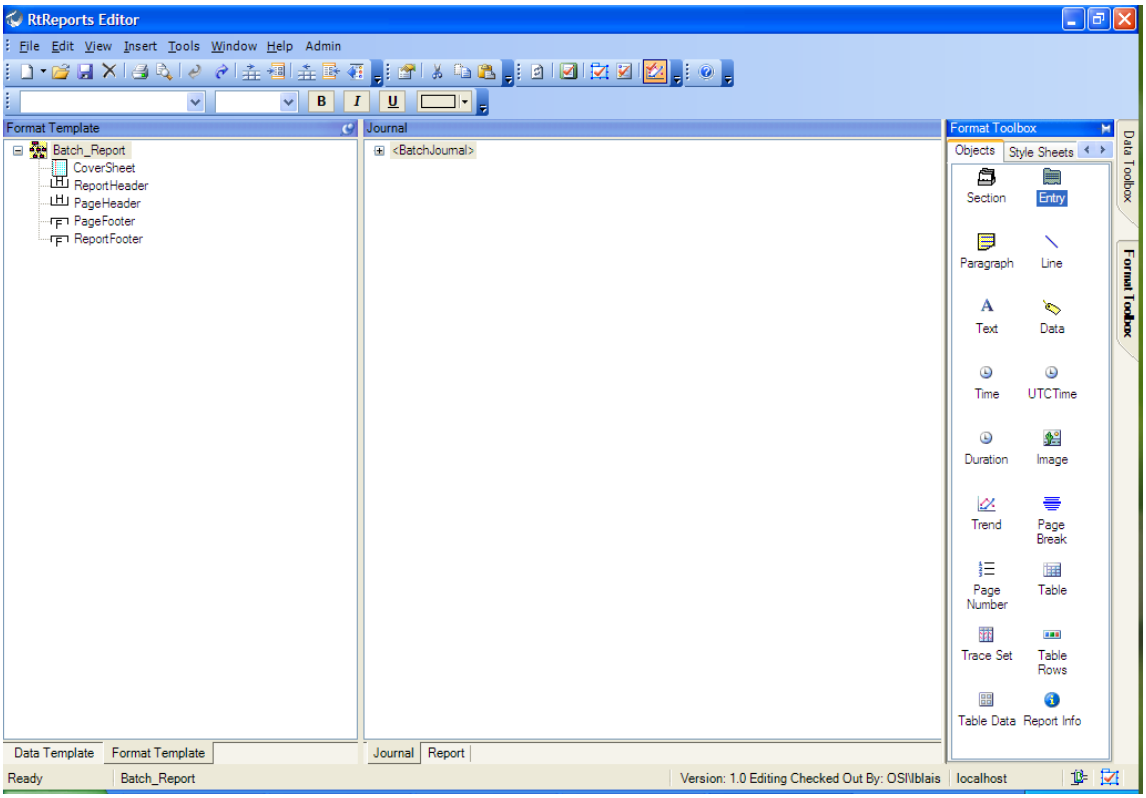
Report Preview

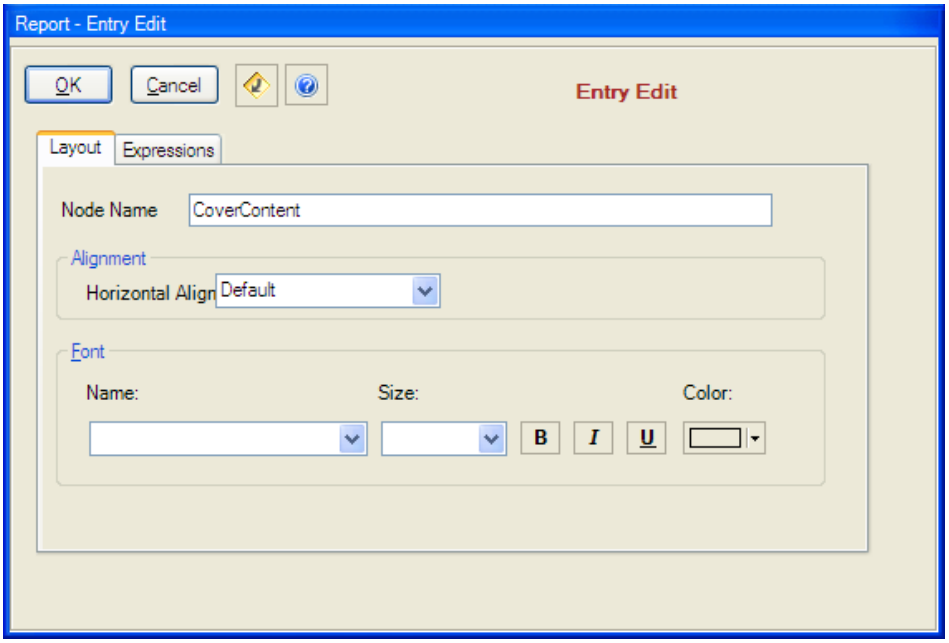


Solution 3: Batch Report Format Template Editing

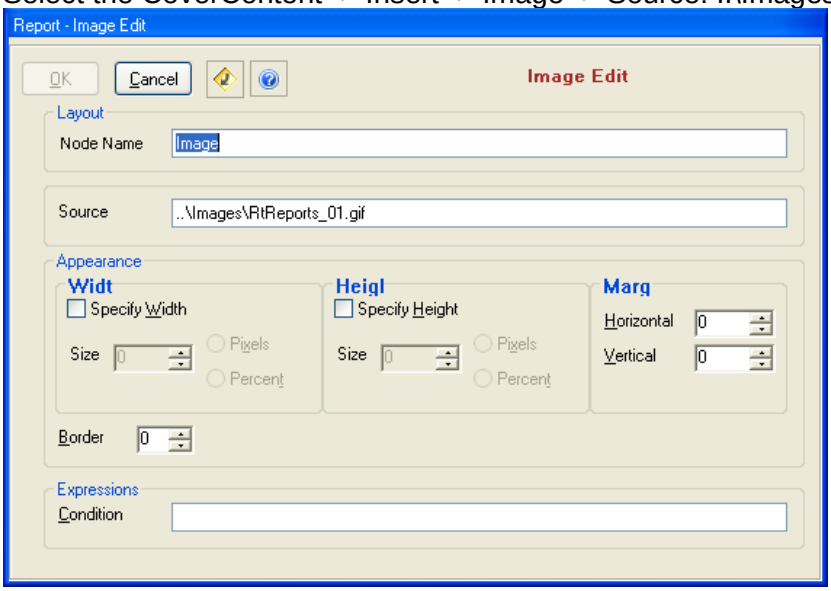
Building *Report Content*

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Open Report the “Batch_Report” template from the previous exercise
3. Select the *Format Template* tab
4. Select the CoverSheet -> insert -> Entry… -> Node Name: CoverContent -> OK

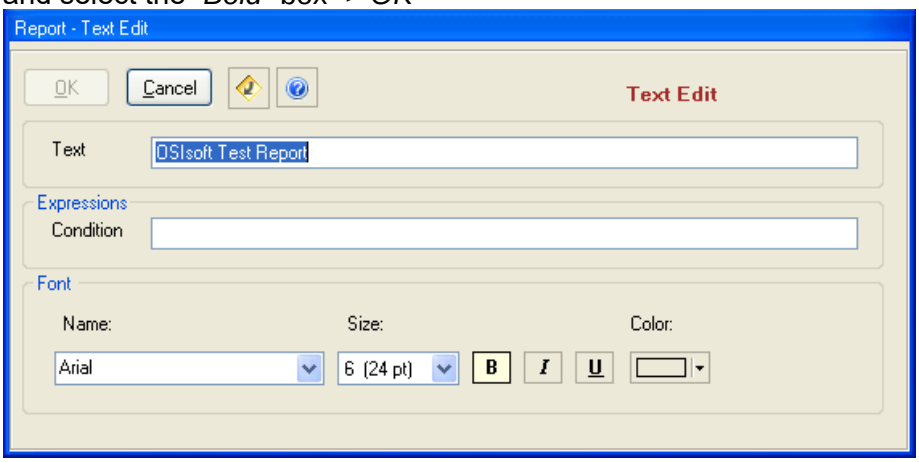




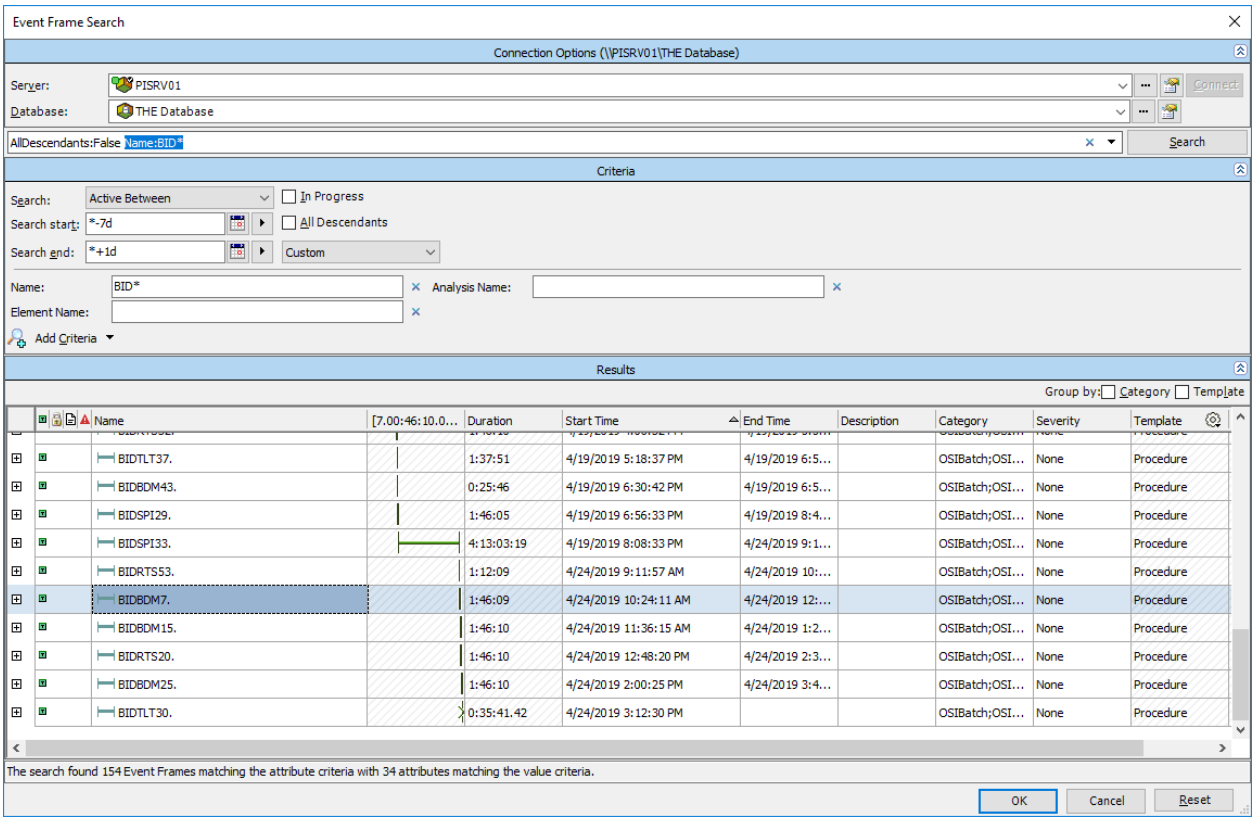
5. Select the CoverContent -> Insert -> Image -> Source: ..\Images\RtReports_01.gif



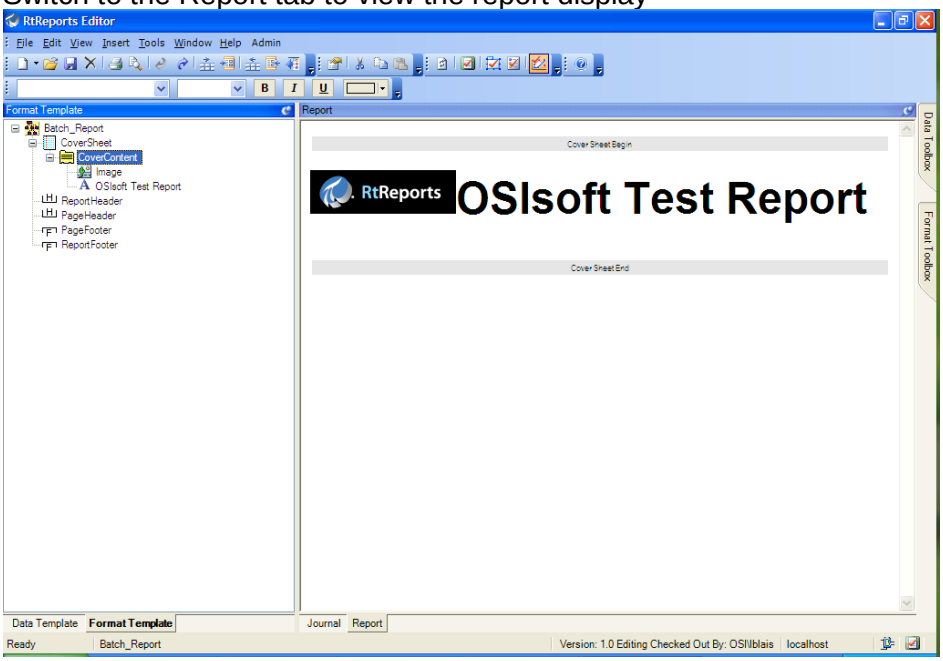
6. CoverContent Entry object -> Insert -> Text object -> Text: OSIsoft Test Report -> *Font* Section, Select *Arial* from *Name* dropdown menu, select 6 from the Size dropdown menu and select the "*Bold"* box -> *OK*



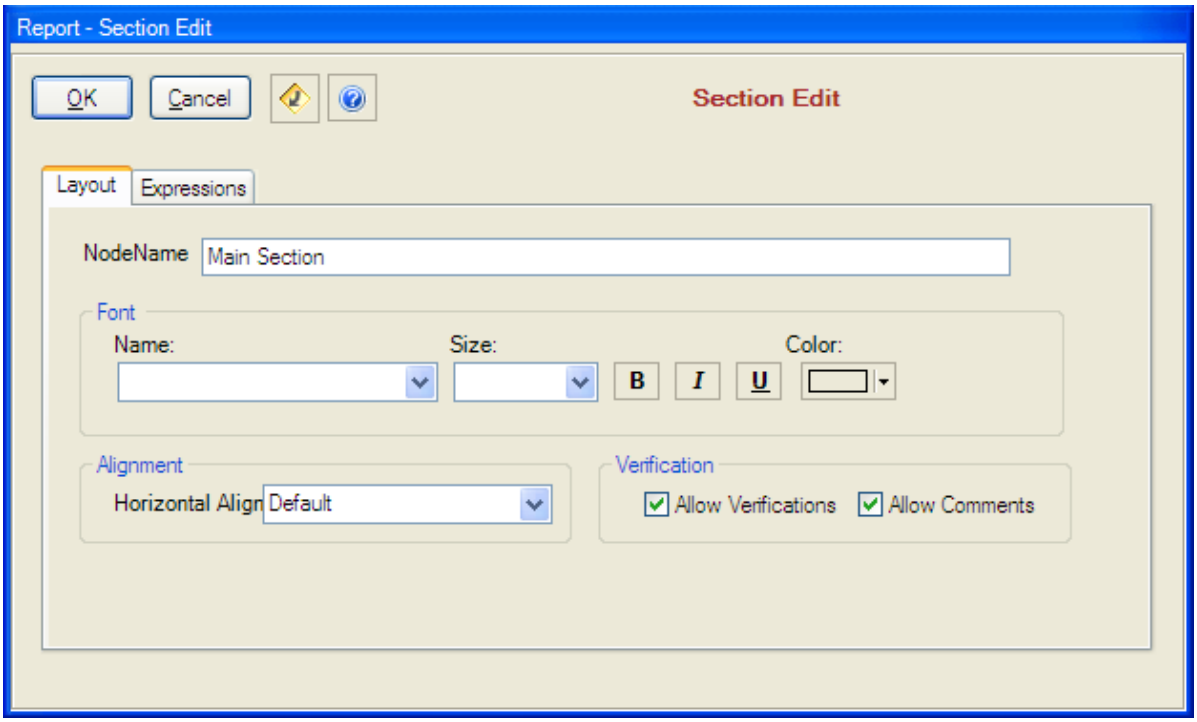
7. To test the Batch_Report: click View menu -> Test Report -> Use Event Frames or press F3 -> in the Event Frame Search prompt, expand Connection Options pane -> verify Server (AF Server) and Database; in the Criteria pane -> edit the Search criteria -> press Search -> in the Results pane, find and select BIDxxx *Procedure* -> OK



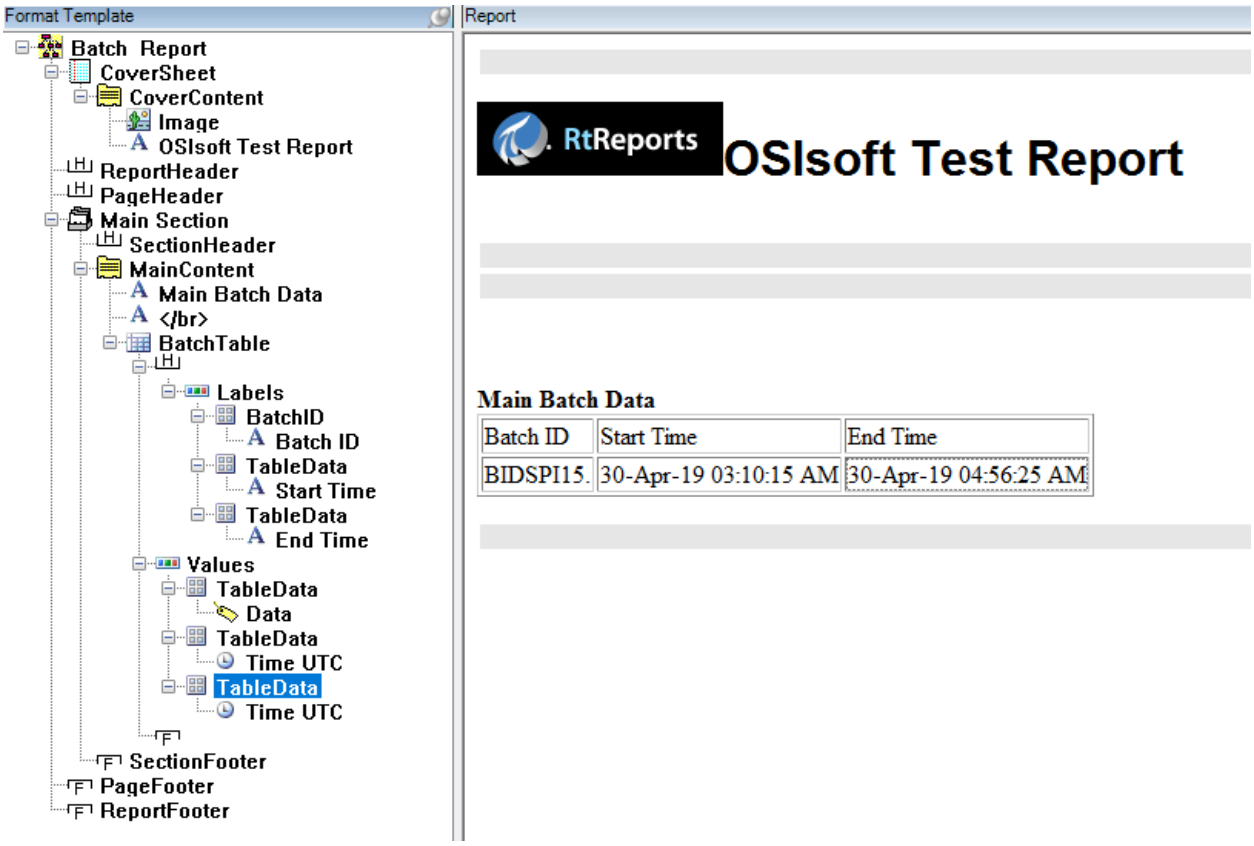
8. Switch to the Report tab to view the report display



9. Back to the Format Template tab, select the root -> Insert -> Section -> Node Name: Main Section -> OK



10. Expand the Main Section -> double clicks Entry -> Node Name: MainContent
11. Select MainContent -> *Insert* -> *Text* -> Text: Main Batch Data
12. MainContent -> *Insert* -> *Table* -> Node Name: BatchTable and set it to include 1 rows and 3 columns -> switch to the *Border* tab -> *Frame*: *Border* -> Border Width: 1 -> *OK*
13. Expand the BatchTable, rename the *TableRow* to *Values*
14. Expand the Values row and rename each *TableData* respectively to *BatchID*, *StartTime* and *EndTime*
15. Expand the Labels row and rename each *TableData* to *BatchID*, *StartTime* and *EndTime*
16. Select the Values row -> Copy -> select table header -> paste -> rename Values to *Labels*
17. Expand the *Labels* section, insert a *Text* object in each child object. Ensure the text content is the same as the node name
18. In the *Values* section, select the BatchID cell -> Insert -> *Data* Object -> in the *Journal* tab, browse the Journal structure, find and drag into the *Source* field the *BatchJournal/BatchID* item; select the StartTime cell -> Insert -> *UTC Time* Object -> find and drag into the *Source* field the *BatchJournal/StartTime/UTCSeconds* item; select the EndTime cell -> Insert -> *UTC Time* Object -> find and drag into the *Source* field the *BatchJournal/EndTime/UTCSeconds* item
19. Save Report Template and Refresh or F5


Format Template
Batch Report
CoverSheet
CoverContent
Image
OSIsoft Test Report
ReportHeader
PageHeader
Main Section
SectionHeader
MainContent
Main Batch Data
</br>
BatchTable
Labels
BatchID
Batch ID
TableData
Start Time
TableData
End Time
Values
TableData
Data
TableData
Time UTC
TableData
Time UTC
SectionFooter
PageFooter
ReportFooter
Report
RtReports
OSIsoft Test Report
Main Batch Data
Batch ID
Start Time
End Time
BIDSPI15.
30-Apr-19 03:10:15 AM
30-Apr-19 04:56:25 AM

Lesson 4: Using Iterations

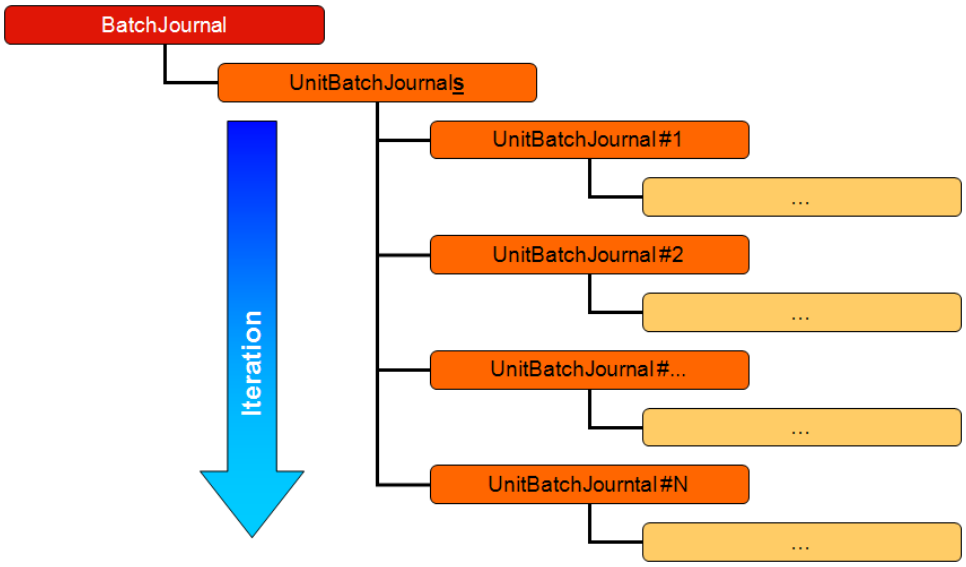
Building reports with simple hierarchies is easy. But what happens if you are building a report that might have many or varying levels? You would not want to hard-code the units even if you knew how many you had. It would be much more efficient if you could simply build your logic once, and tell the system to simply build a section or table row for each item in the list of results it encounters.

This is called iteration.

Working with Iterator Expressions

Most reports are repetitive by nature, such as production summary reports, clean utility reports, computer usage reports, etc. Often, you will want to show the same format for each phase on a report, or the same format for each hour of clean utility usage, etc.

To loop on a certain collection in the Journal, you can use an Iterator expression. An Iterator expression uses the same syntax as the Source Expression, but it controls the current context of the Source Expressions that are used.



In this example, we are showing a BatchJournal with its own UnitBatchJournal. Typically, the UnitBatchJournals will hold one or many UnitBatchJournal, one for

each PIUnitBatch under the PIBatch. Very often, the number of Journals included in a Journal is unknown. Nevertheless, we still need to process all the entries.

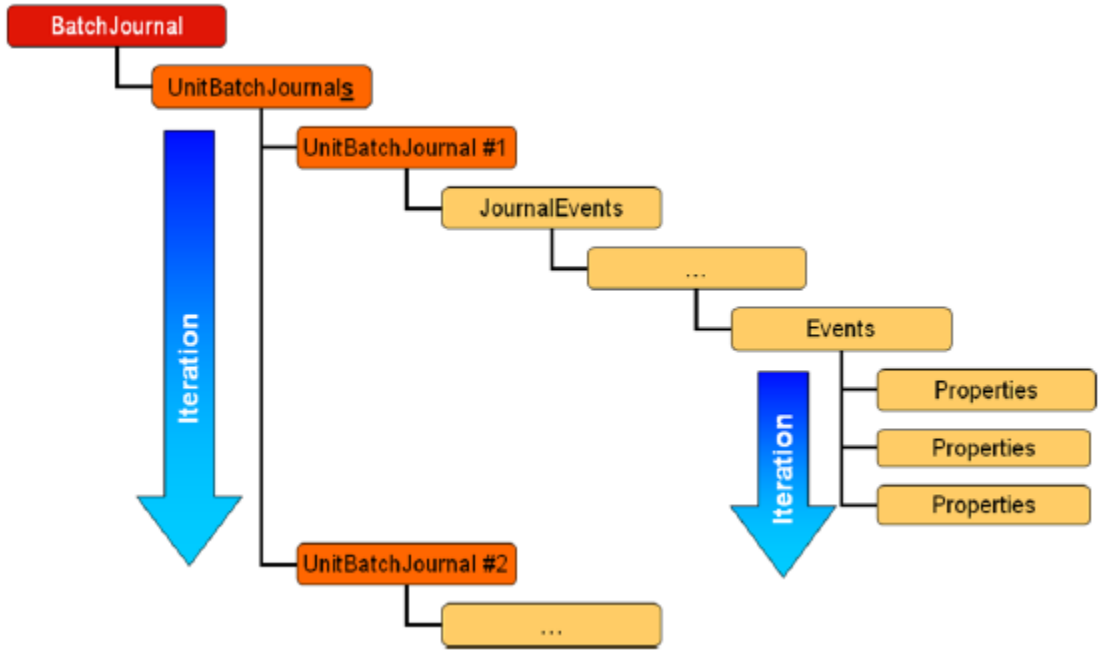
Where is the Iterator Expression?

Using the formatting elements, you will notice that many of them have an Expressions tab. If you need to iterate on a portion of your journal, you must enter in the Iterator field the element on which the iteration process will take place.

Multi-Level Iterators

Multi-Level Iterators

Many scenarios will require that you use multi-level Iterators. By multi-level we mean that iteration will take place within iteration. As an example, we created a business rule that retrieves the compressed values for a tag based on the PIUnitBatch running. This rule will yield to two sets of results, one for PIUnitBatch #1 and one for PIUnitBatch #2. If we want to display the information for both PIUnitBatches, we will create a first iterator on the UnitBatchJournal. Then, we will iterate on the Properties of the JournalEvents of each PIUnitBatch.



One Example:

Complete path to the data:

```
BatchJournal/UnitBatchJournals/UnitBatchJournal[Name="R-
2401"]/JournalEvents/JournalEvent[Name="CompressData"]/TerminalStates/TerminalSta
te[StateName="Direct
Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="GetCompr
```

```
essData"]/Value/TraceSet[@Name="GetCompressData.Tag"]/Events/Properties[position(
)=1]/Property[@Descriptor="22000"]
```

Iterator on PI UnitBatches

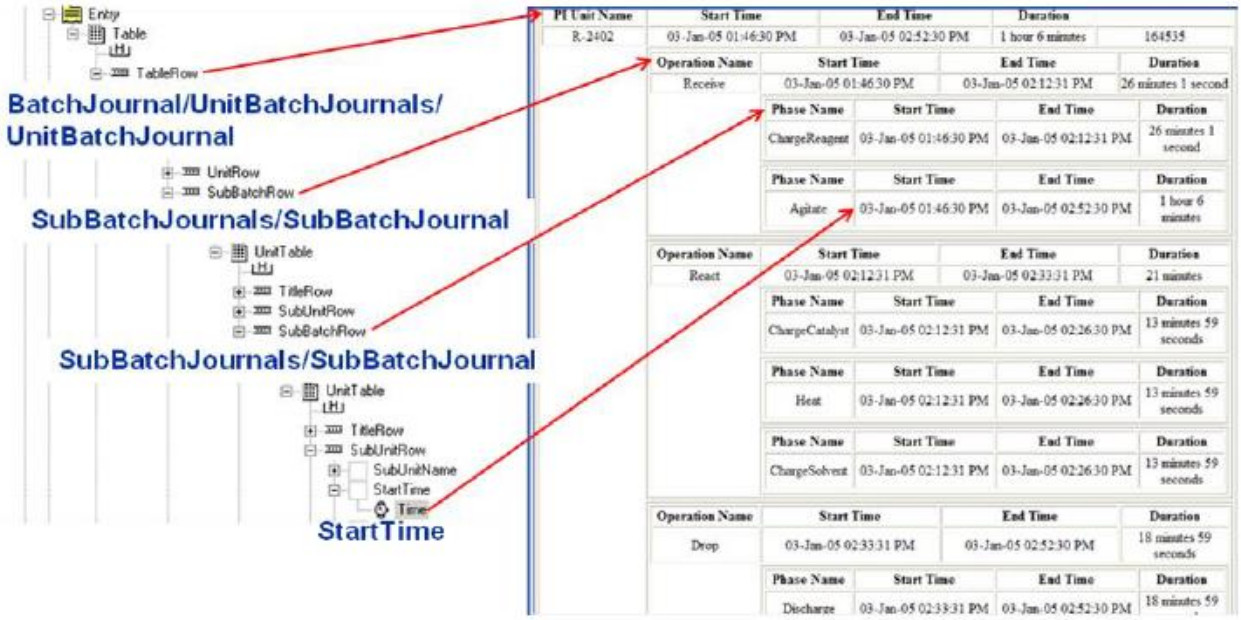
```
BatchJournal/UnitBatchJournals/UnitBatchJournal
```

Iterator on the Events:

```
JournalEvents/JournalEvent[Name="CompressData"]/TerminalStates/TerminalState[Sta
teName="Direct
Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="GetComp
ressData"]/Value/TraceSet[@Name="GetCompressData.Tag"]/Events/Properties
```

Actual Value:

```
Property[@Descriptor="22000"]
```



Exercise 4: Adding Iterators to the Format template

Objectives

Learn how to use iterator to display multiple instances of repeated objects in a report.

Problem Description

Modify the *Batch_Report* from the previous Exercise to display the Unit-Procedures data (Unit name, Start Time, and End Time) in a table that consist of a Label row and Values row.

Approach

Insert an Entry “IterativeContent” under the “Main Section” section.

Insert a “UnitBatchTable” table under the IterativeContent entry

Identify that there are multiple PIUnitBatches.

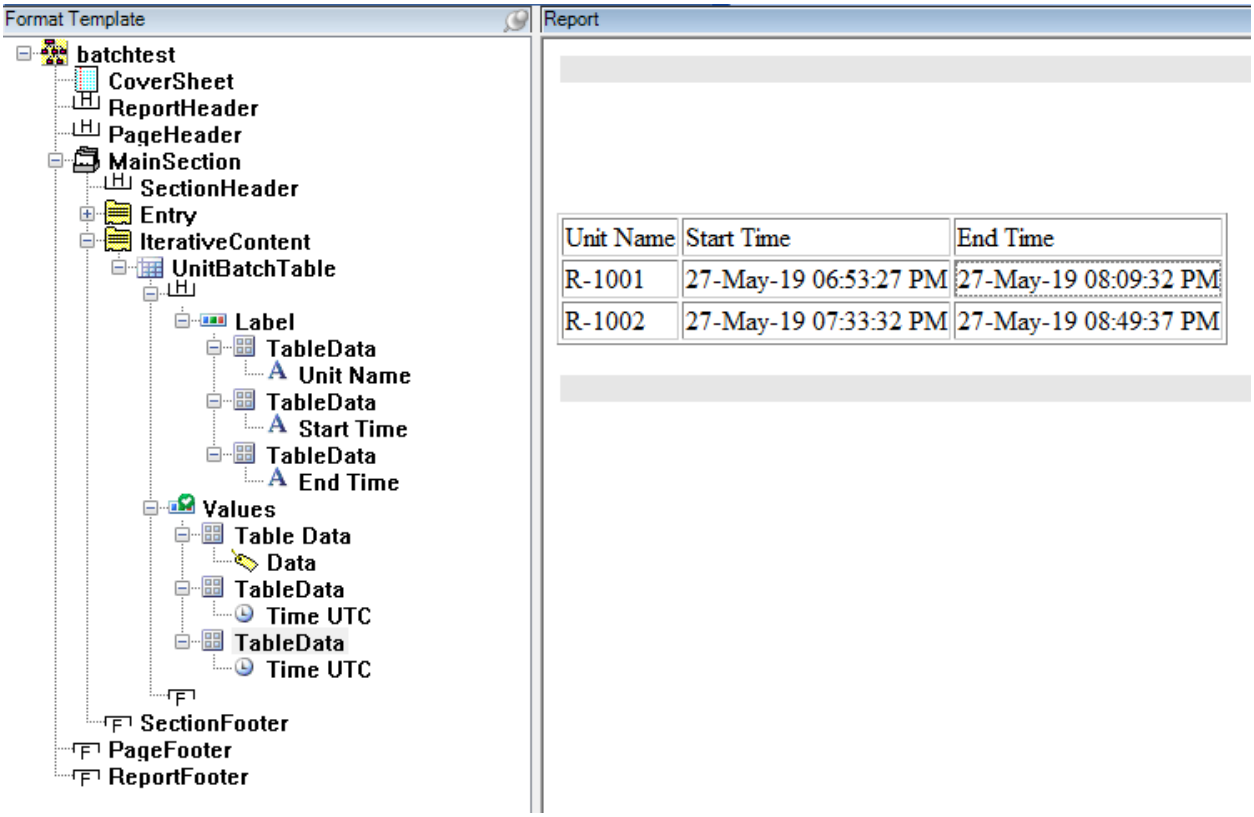
Add Iterator to the Values row of the table at the PIUnitBatches

Note: the *iterator* xml path must not be repeated in any of the children objects (remove manually if necessary).

Example: to display the Unit Name of each Unit-Procedure (xml path: *BatchJournal/UnitBatchJournals/UnitBatchJournal/Name*):

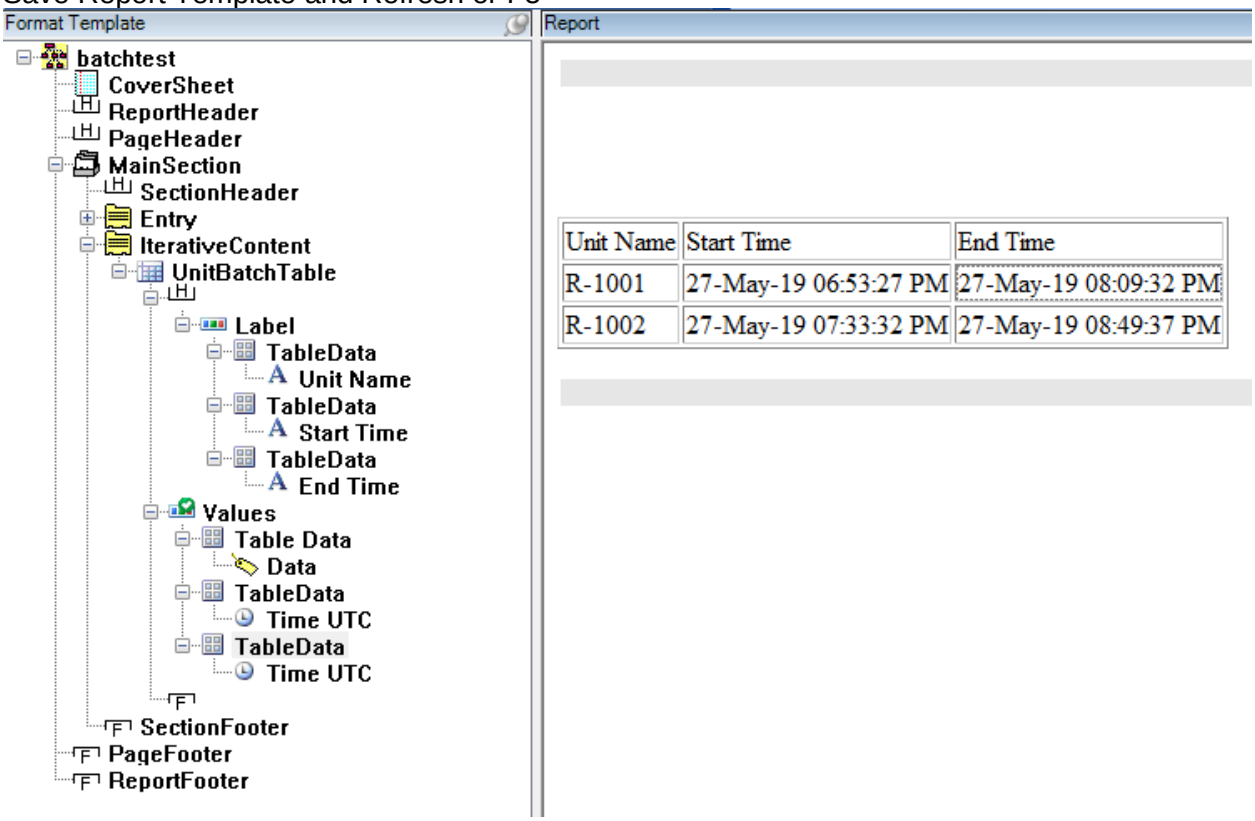
- The xml path for the table row *Iterator*: *BatchJournal/UnitBatchJournals/UnitBatchJournal*
- The xml path Unit cell: *Name*

Report Preview



Solution 4: Adding Iterators to the Format Template

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Open the Batch_Report report template from the previous exercises
3. Select Main Section -> Insert -> Entry -> Node Name: IterativeContent -> OK
4. Select IterativeContent -> Insert -> Table -> Node Name: UnitBatchTable -> Rows: 1 Columns: 3 -> switch to the Border tab -> Frame: Border -> Border Width: 1 -> OK.
5. Expand the UnitBatchTable -> rename the TableRow to *Values*
6. Expand the Values row and rename each TableData to UnitName, StartTime, and EndTime
7. Select the Values row -> copy -> select the table header -> paste -> rename the row to *Labels*
8. Expand the Labels row, select each cell -> Insert -> Text -> Text: same as the node name
9. Double-click the Values row -> *Expressions* tab -> switch to the *Journal* tab, find and drag *BatchJournal/UnitBatchJournals/UnitBatchJournal* item in the *iterator* field -> OK. **Notice** a green checkmark on the Values node
10. Expand the Values row -> select the UnitName cell -> Insert -> Data object -> find and drag to Source the *Name* item of the UnitBatchJournal; select the StartTime cell -> Insert -> UTC Time object -> find and drag to Source the *StartTime/UTCSeconds* item of the UnitBatchJournal; select the EndTime cell -> Insert -> UTC Time object -> find and drag to Source the *EndTime/UTCSeconds* item of the UnitBatchJournal
11. Save Report Template and Refresh or F5



Lesson 5: Journal Actions

A Journal Action provides the basis for including time series data within a report. It is a set of business rules with compound logic to produce a calculated result from raw data.

A Journal Action can take one of two paths to get data:

- **Action Results** - which use the start and end time of the parent context and retrieve Result Definitions
- **Action States** - which means defining a series of start and end times to retrieve Result Definitions.

Journal actions produce a Journal entry upon a change of state. A state is defined by a Transition Equation formulated using the PI Performance Equations (PI PE) syntax and functions. For every state there is one Transition Equation returning the number of the new state. Every time a new state is entered, a new set of result expressions is evaluated.

Variables and placeholders are called **Action Parameters** in RtReports. You may use Action Parameters because of their data types that launch intuitive search controls for finding data within their native data stores.

Context

Batch reports will get the Context Start and Context End times from the context that the user selects. This means that a Journal Action at the Unit Batch level will use the start and end time of the Unit Batch as the Context.

For time-based reports the bounding contexts will be the start and end time of the defined iteration period. So an hourly report will select the start and end of the hour for the Context Start Time and End Time.

Make sure when you insert your Journal Actions you inset it at the right level.

Action Parameters and Results

The following parameter types are supported:

Literal	The value column is considered a static numeric or string data type depending on the contents entered
PITag	The value column is considered a PI Tag
PIProperty	The value column is considered a PIProperty

PIAlias	The value column is considered a PI Alias. There are 2 types of aliases supported in RtReports, Absolute Aliases and Unit Relative Aliases
AFAttribute	The value column is considered an AF Attribute. There are 2 types of AF Attribute supported in RtReports, Absolute Path and Relative Path
EFAttribute	The value column is considered an EF Attribute

Depending on what type you chose, a wizard could be invoked.

The Dynamic column is used to signify whether the parameter value should be resolved before executing the Action States and Result Definitions.

One Example: Configuring a Trend for the Report

Configuring a Trend Result Definition will output to the Journal the information required by the Trend format object to display a trend in a report.

Profile (GetProfile) Value Result Definition

The Profile Result Definition will retrieve the **Start Value**, **End Value**, **Maximum Value**, and **Minimum Value** of a single or multiple Tags between the Context Start and Context End. The settings for the Profile result are defined below:

DataSource = the PI Tag/Alias to retrieve Compressed Values.

Summary (GetSummary) Value Result Definition

The Summary Result Definition will retrieve Summary values for a PI tag. The supported summary values are **Minimum**, **Maximum**, **Total**, **Average**, **Standard Deviation**, **Range** and **Mean**. These are similar to the Dataset out put in PI ProcessBook and RtWebParts or Calculated Data in PI DataLink.

TraceSet

Delivers results in tabular format.

Exercise 5: Adding a Journal Action

Objectives

Discover *Trends*, *Summary,* and *Profile Result Definitions* to *Journal Action.*

Problem Description

Add a *Journal Action* that consists of a *Trend, Summary Value (of Minimum, Maximum, and Average)*, and *Profile (Values at the Start Time and End Time) Result Definitions* of the *PITag CDT158* of a reactor under the Batch level to the existing Batch_Report report template.

Display those Result Definitions.

Approach

Insert a Journal Action to the Batch level

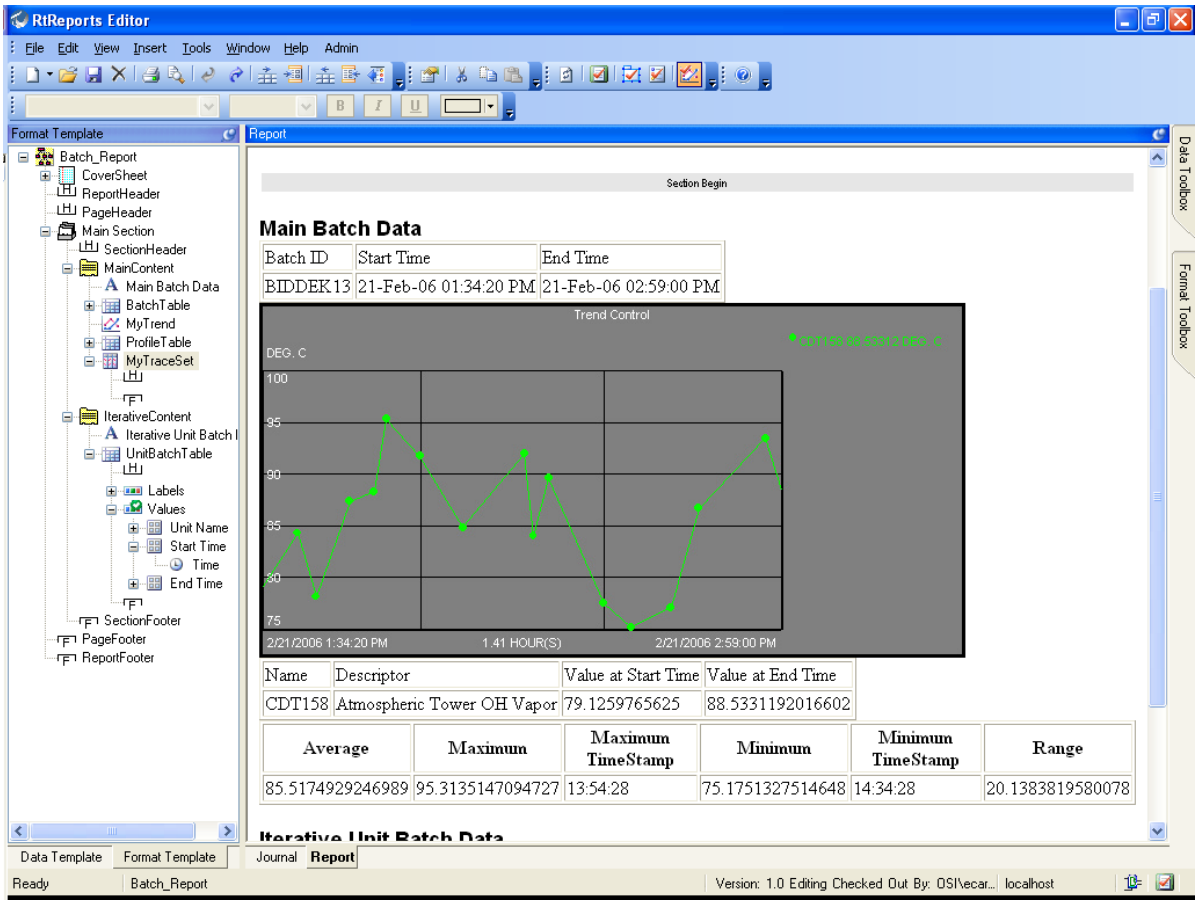
Add Result Definitions of Trend, Summary Value, and Profile for tag CDT158

Use the *Trend* object to trace the value of CDT158 over the Batch duration.

Use the *Trace Set* object to display the *Summary Value* values

Use the *Table* object to display the *Profile* values

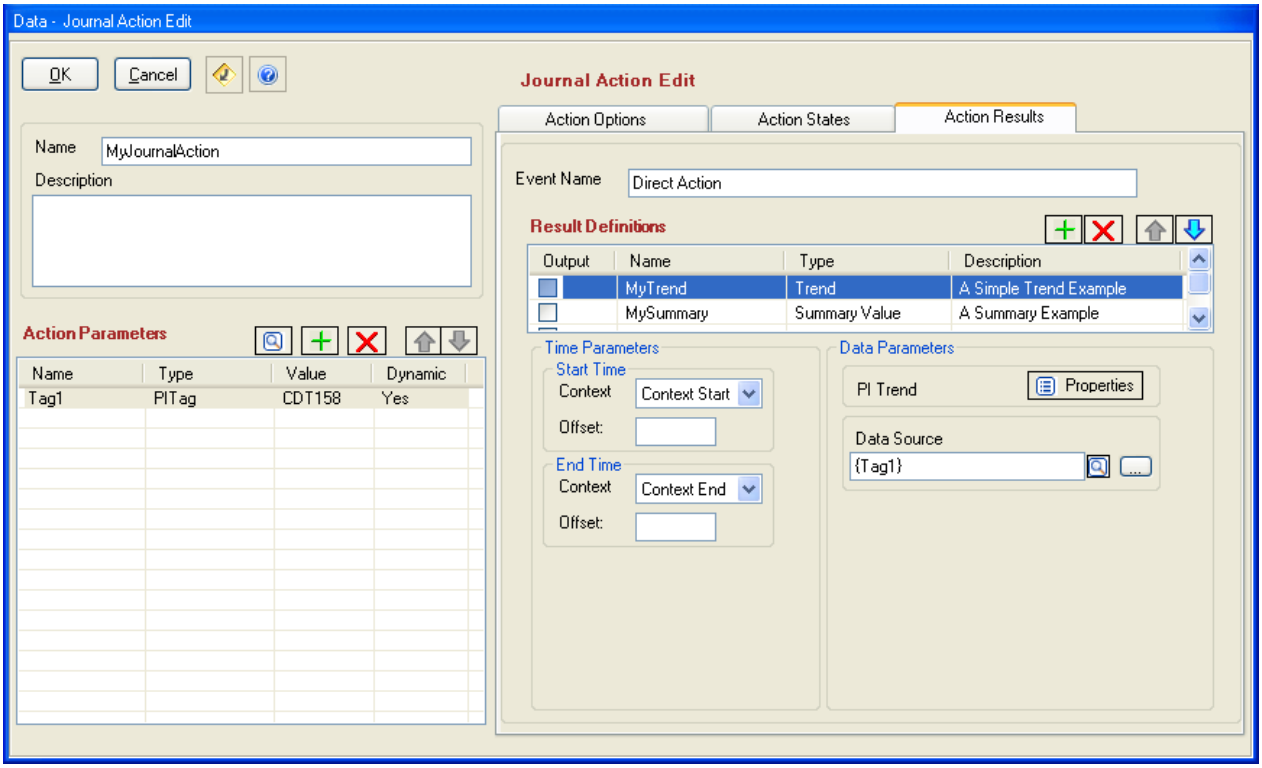
Report Preview



Solution 5: Adding a Journal Action

Adding *Journal Action* to *Data Template*

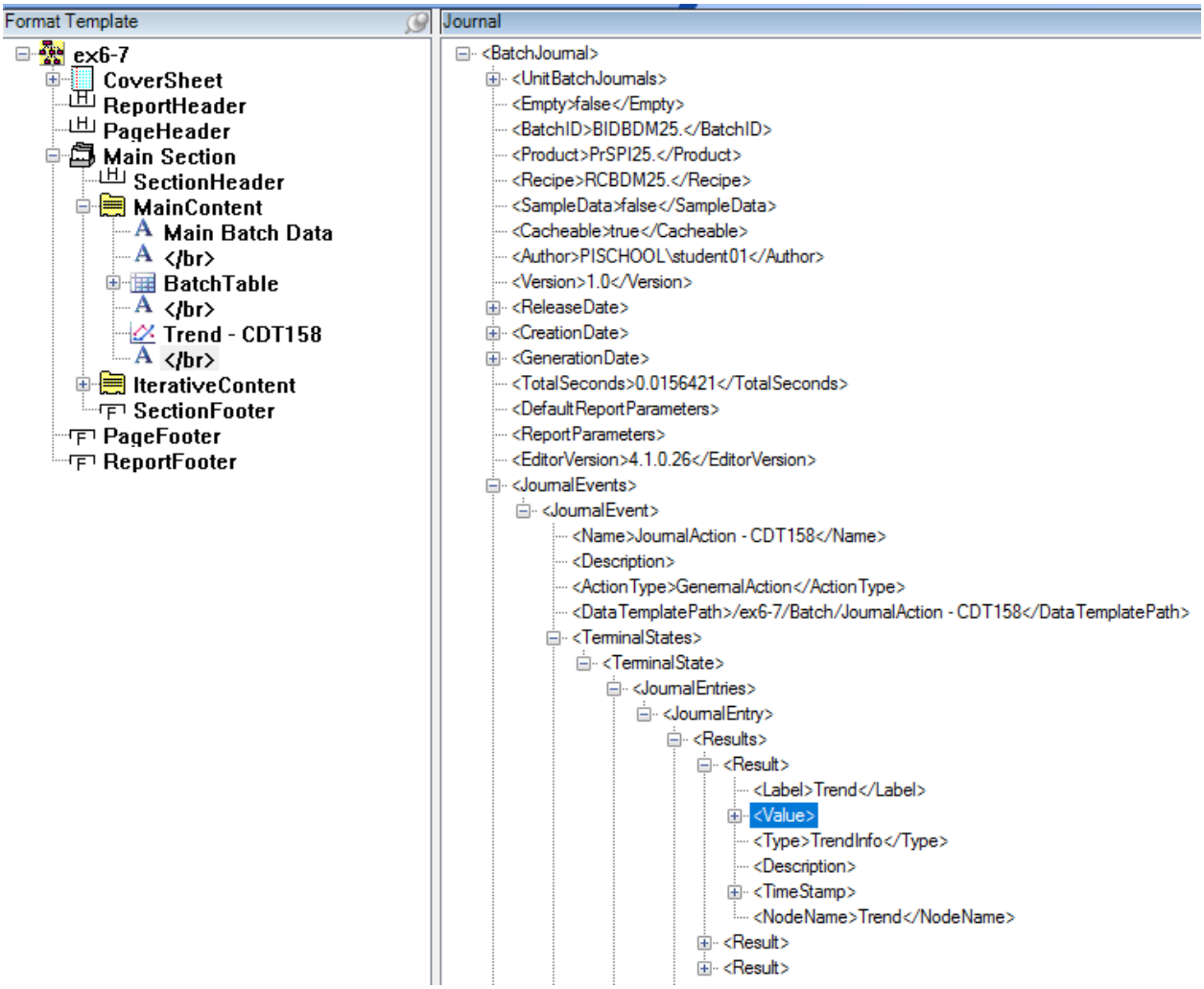
1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Open the existing Batch_Report template
3. In the Data Template tab, select Batch -> Insert -> Journal Action -> Name: "JournalAction – CDT158"
4. Continue, under *Action Parameters* -> green **+** -> Name: Tag -> Type: PITag -> browse tag CDT158 -> OK -> under *Journal Action Edit* -> *Action Results* tab…
5. -> green **+** -> Name: Trend -> Type: Trend -> drag "Tag" (left) to Data source (right). Notice: { }
6. -> green **+** -> Name: Summ -> Type: *Summary Value* -> drag "Tag" (left) to Data source (right) -> check "Yes" for Minimum, Maximum, and Average
7. -> green **+** -> Name: Prof -> Type: *Profile* -> drag "Tag" (left) to Data source (right) -> check "Yes" for Start and End -> OK



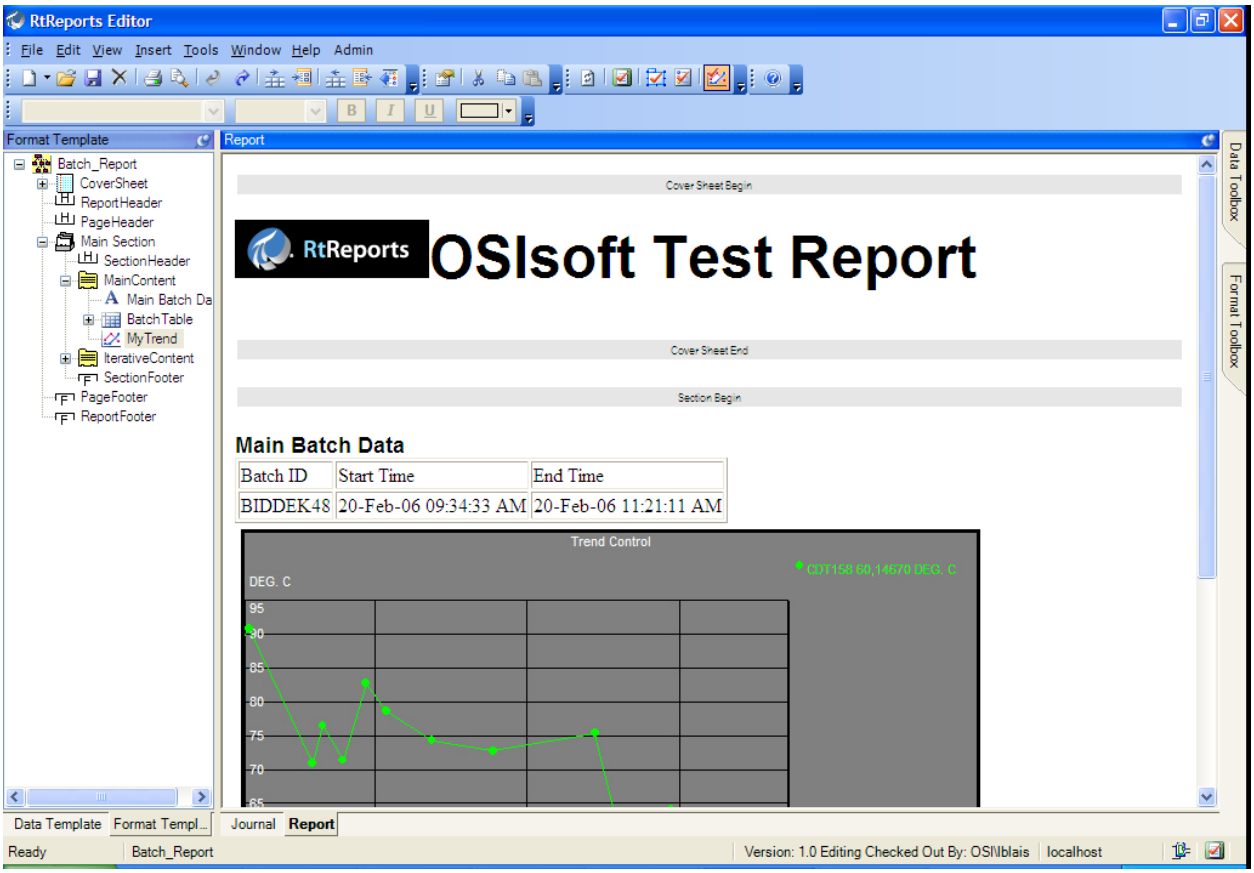
8. Save Report Template -> F3 to test or F5 to refresh

Displaying the Trend

1. Switch to the Format Template tab -> expand the “Main Section”
2. Select MainContent -> Insert -> Trend object -> Node Name: Trend – CDT158 -> browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="JournalAction - CDT158"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Trend"]/Value* to the Source field



3. Save Report Template and Refresh or F5



Adding a TraceSet (smart table) for the Summary Data

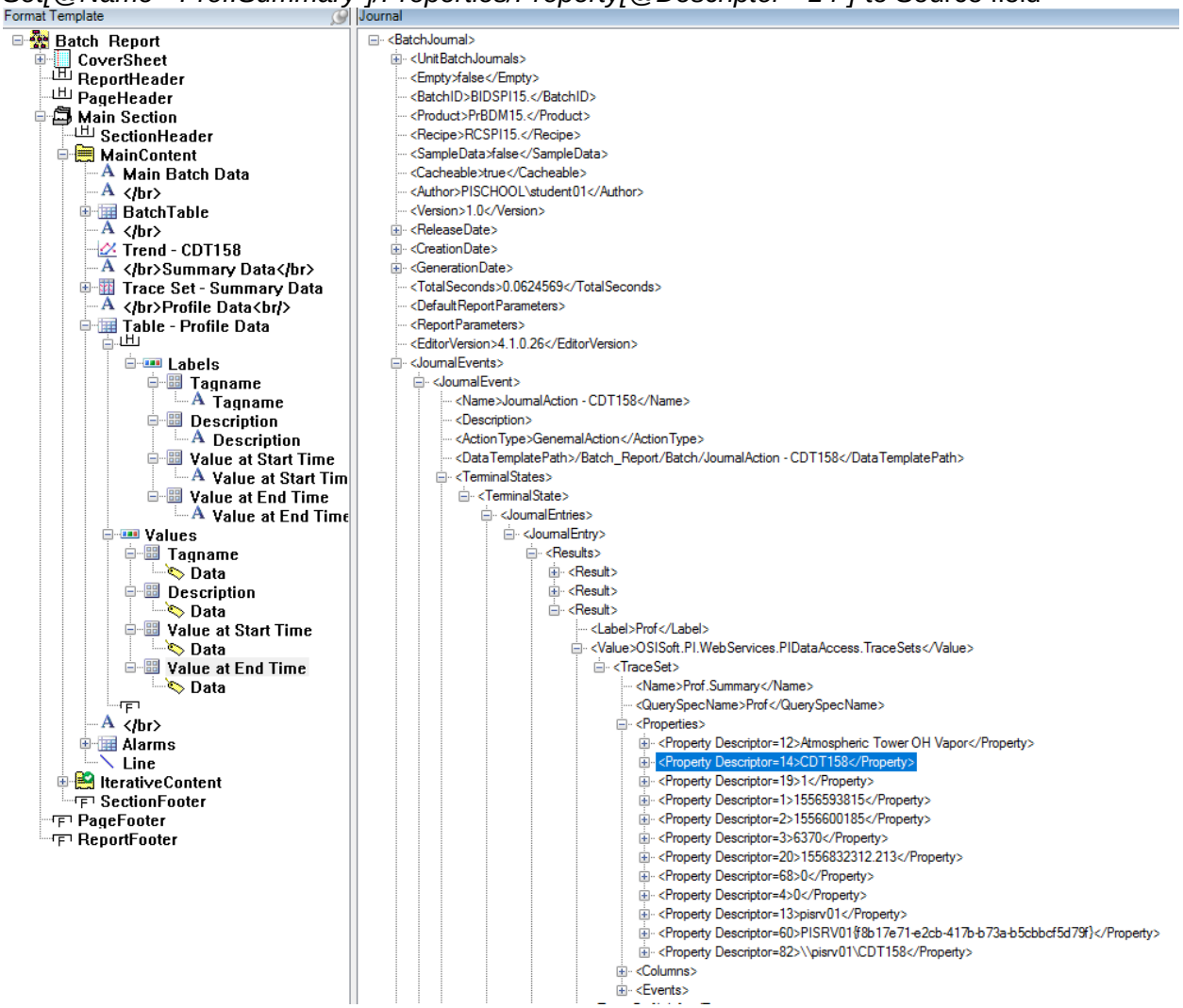
1. In Format Template tab -> expand the “Main Section”
2. Select “MainContent” -> Insert -> *Trace Set* -> Node Name: Trace Set – Summary -> browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="JournalAction - CDT158"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Summ"]/Value/TraceSet[@Name="Summ.Summary"]* to Source field -> Rows Definition tab -> red **X** to remove columns *Time, ErrorCode, ErrorDescription, and IsGood* -> Border tab -> Frame: Border -> Border Width: 1 -> OK
3. Save Report Template -> Refresh or F5

Adding a Table for the Profile Data

1. In Format Template tab -> expand the “Main Section”
2. Select “MainContent” -> Insert -> Table -> Node Name: Table – Profile Data -> Rows: 1 -> Column: 4 -> Border tab -> Frame: Border -> Border Width: 1 -> OK
3. Expand the “Table – Table -> rename TableRow to *Values*
4. Expand the *Values* row -> Column 1 – Column 4 -> Node Name: Tag, Description, Value at

Start Time, and Value at End Time

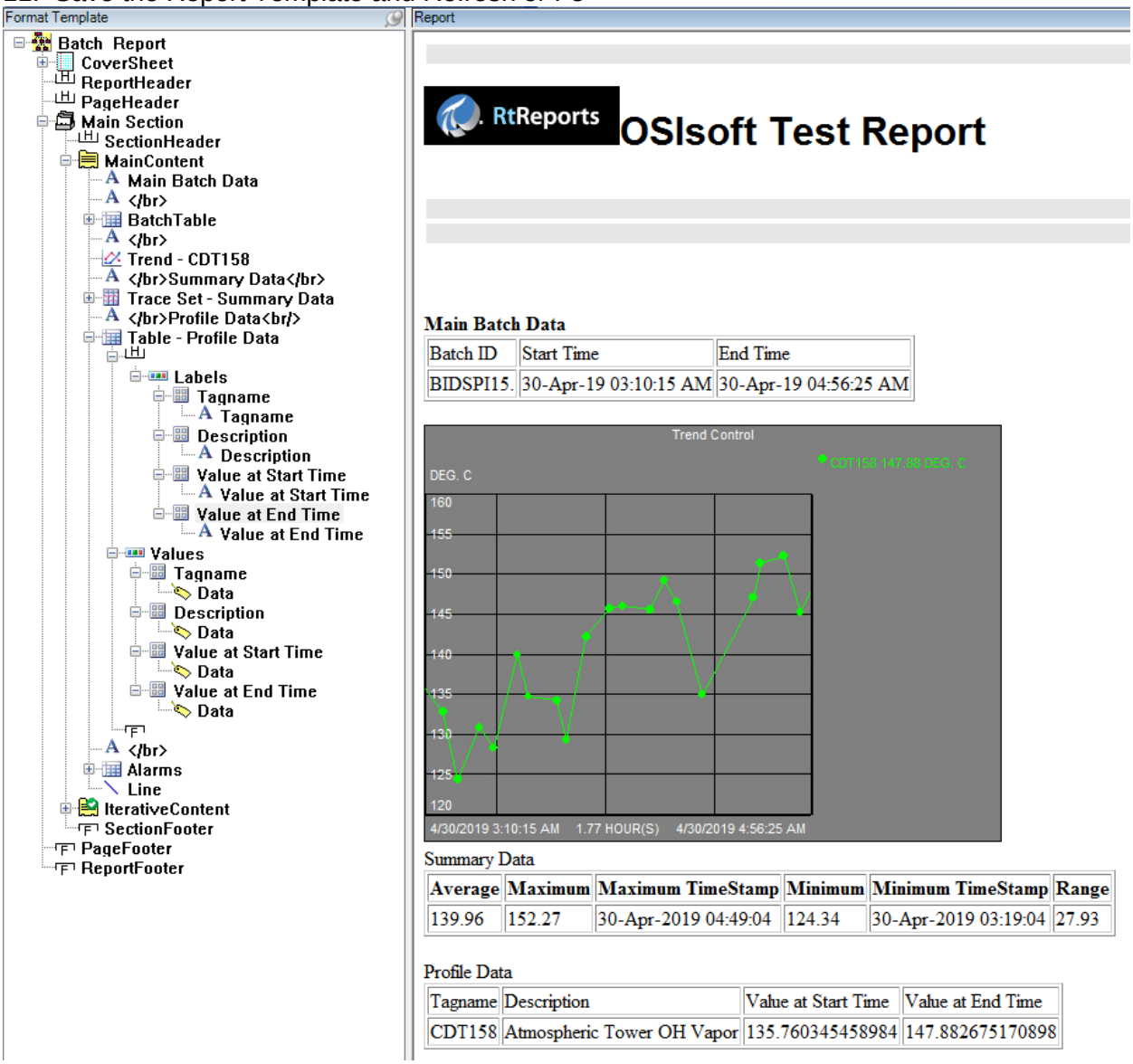
5. Select the *Values* row -> copy -> select the table header -> paste -> rename the row to *Labels*
6. Expand the table header row *Labels* -> select each TableData -> Insert -> Text -> Text: same as column name
7. Expand the Value row -> select the *Tagname* TableData -> Insert -> Data -> browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="JournalAction - CDT158"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Prof"]/Value/TraceSet[@Name="Prof.Summary"]/Properties/Property[@Descriptor="14"]* to Source field



8. Select the *Description* TableData -> Insert -> Data -> browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="JournalAction - CDT158"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Prof"]/Value/TraceSet[@Name="Prof.Summary"]/Properties/Property[@Descriptor="12"]* to the Source field
9. Select the *Value at Start Time* TableData -> Insert -> Data -> browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="JournalAction - CDT158"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Prof"]/Value/Trace*

Set[@Name="Prof.Summary"]/Events/Properties[position()=1]/Property[@Descriptor="22008"] to Source field

10. Select the *Value at End Time* TableData -> Insert -> Data -> browse and drag BatchJournal/JournalEvents/JournalEvent[Name="JournalAction - CDT158"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Prof"]/Value/TraceSet[@Name="Prof.Summary"]/Events/Properties[position()=1]/Property[@Descriptor="22009"] to Source field
11. Save the Report Template and Refresh or F5



Exercise 6: Add Compressed Data

Objectives

Display archived values of an element attribute for the Unit-Procedure duration in a table using iterator.

Problem Description

Display in a table format the *Result Definitions* of Type *Compressed Values* of the *Temp AFAttribute* of each 1K Reactors for the duration of the Unit-Procedure.

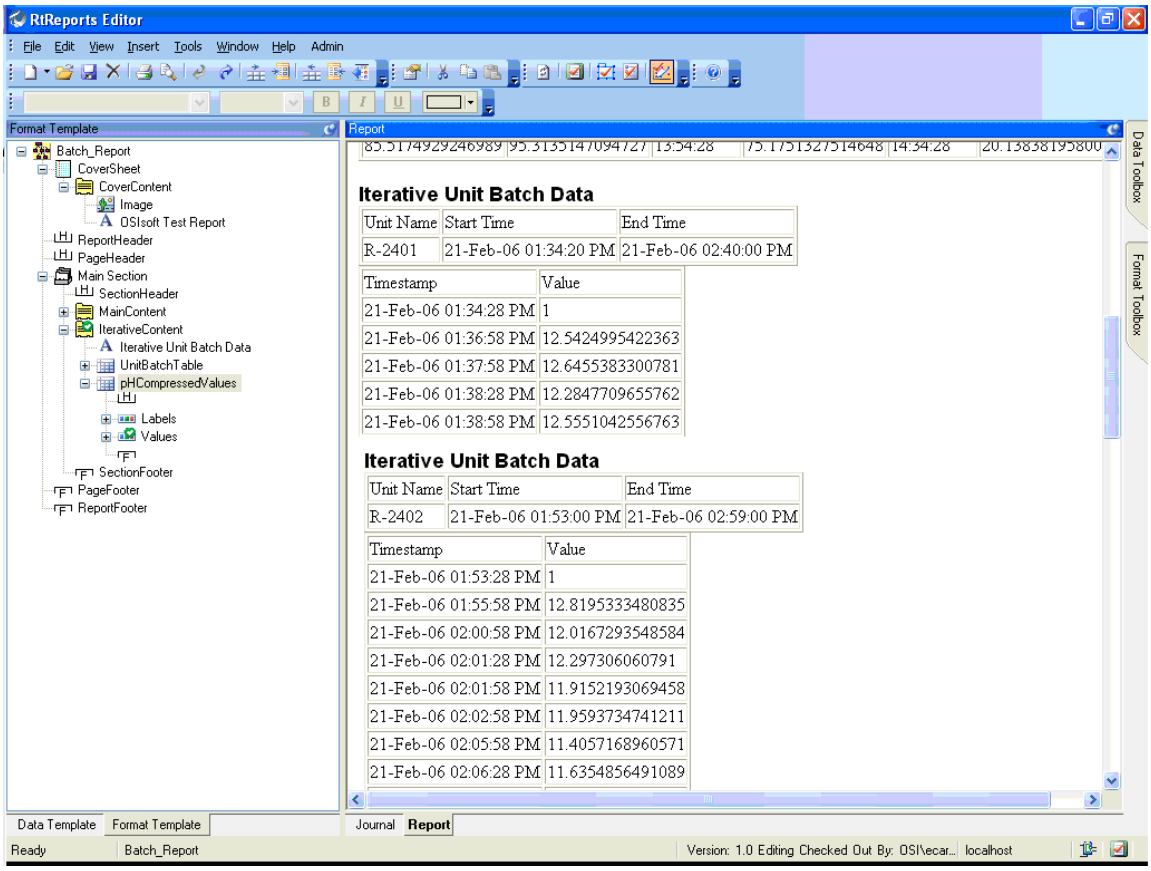
Approach

Add a Journal Action of Result Definitions Compressed Values for the *Temp* element attribute of each 1K Reactors.

Insert a table under the “IterativeContent” entry of the “Main Section” Section to display Timestamp and Value of each archive record.

Suggestion: iterate the “IterativeContent” entry at the *UnitBatches*

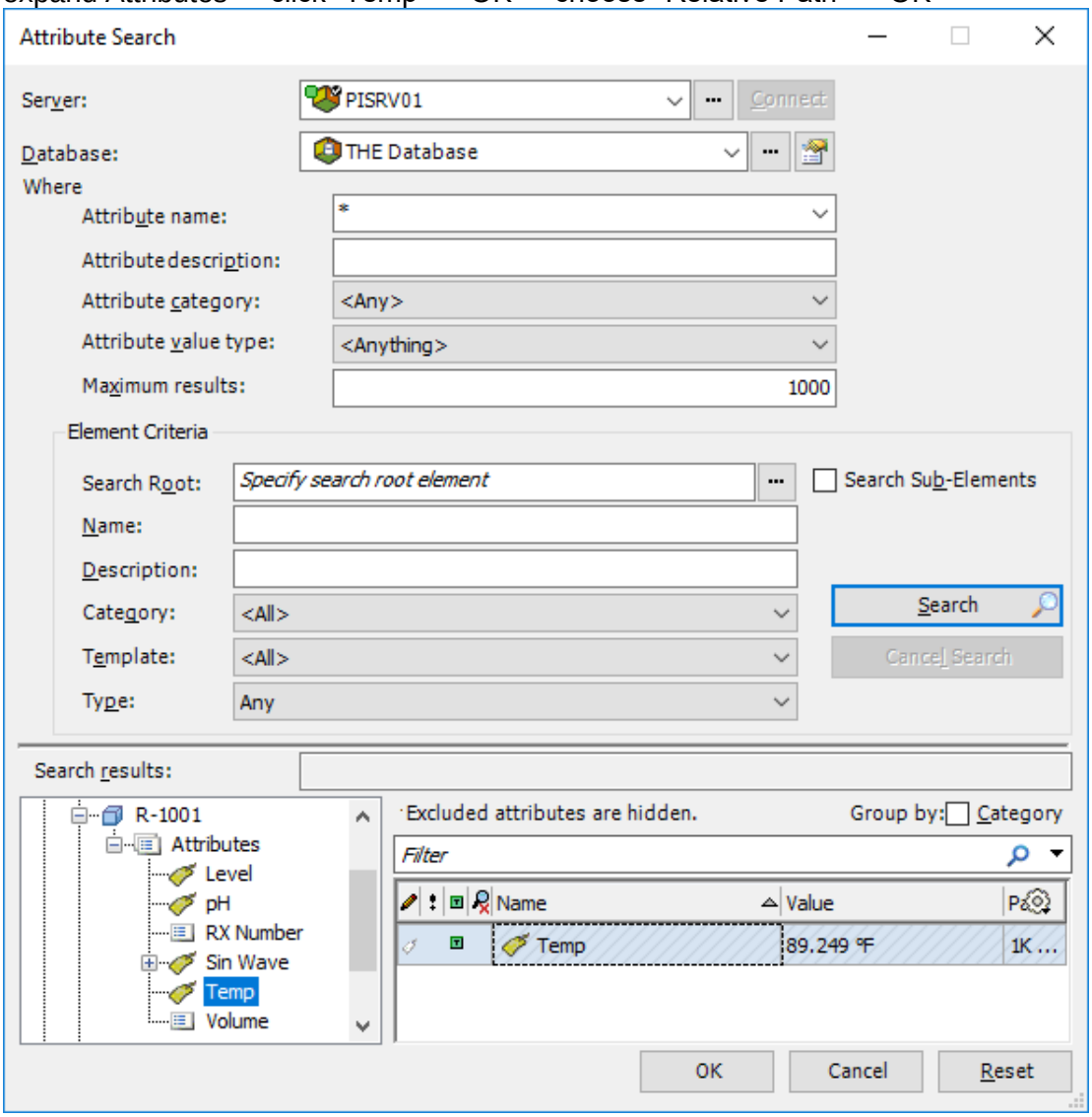
Report Preview



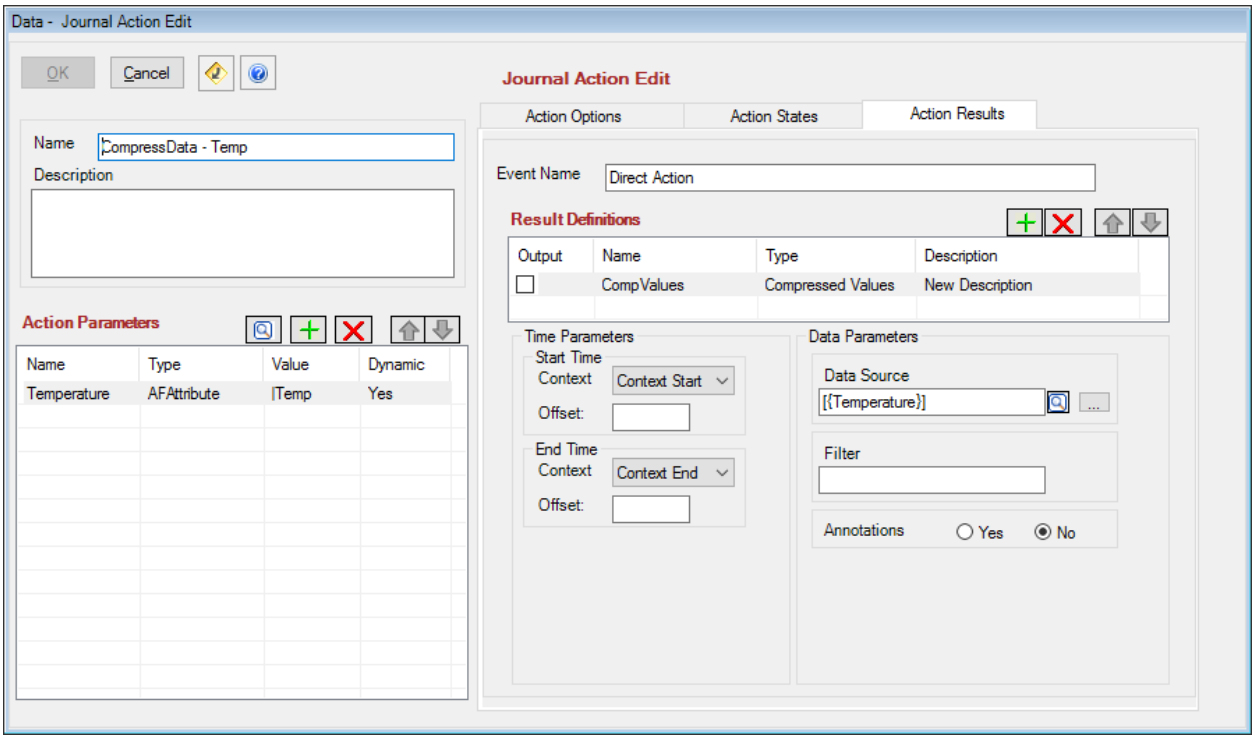
Solution 6: Add Compressed Data

Create a Journal Action for the extraction of the archived data

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Open the *Batch_Report* template for previous exercise
3. Select the Data Template tab
4. Expand Batch
5. Select the UnitBatch -> Insert -> Journal Action -> Name: CompressData – Temp -> …
6. Action Parameter -> + -> Name: Temperature -> Type: AFAttribute -> Server: PISRV01 -> Database: THE Database -> expand 1K Reactors (left-bottom) -> expand R-1001 -> expand Attributes -> click “Temp” -> OK -> choose “Relative Path” -> OK



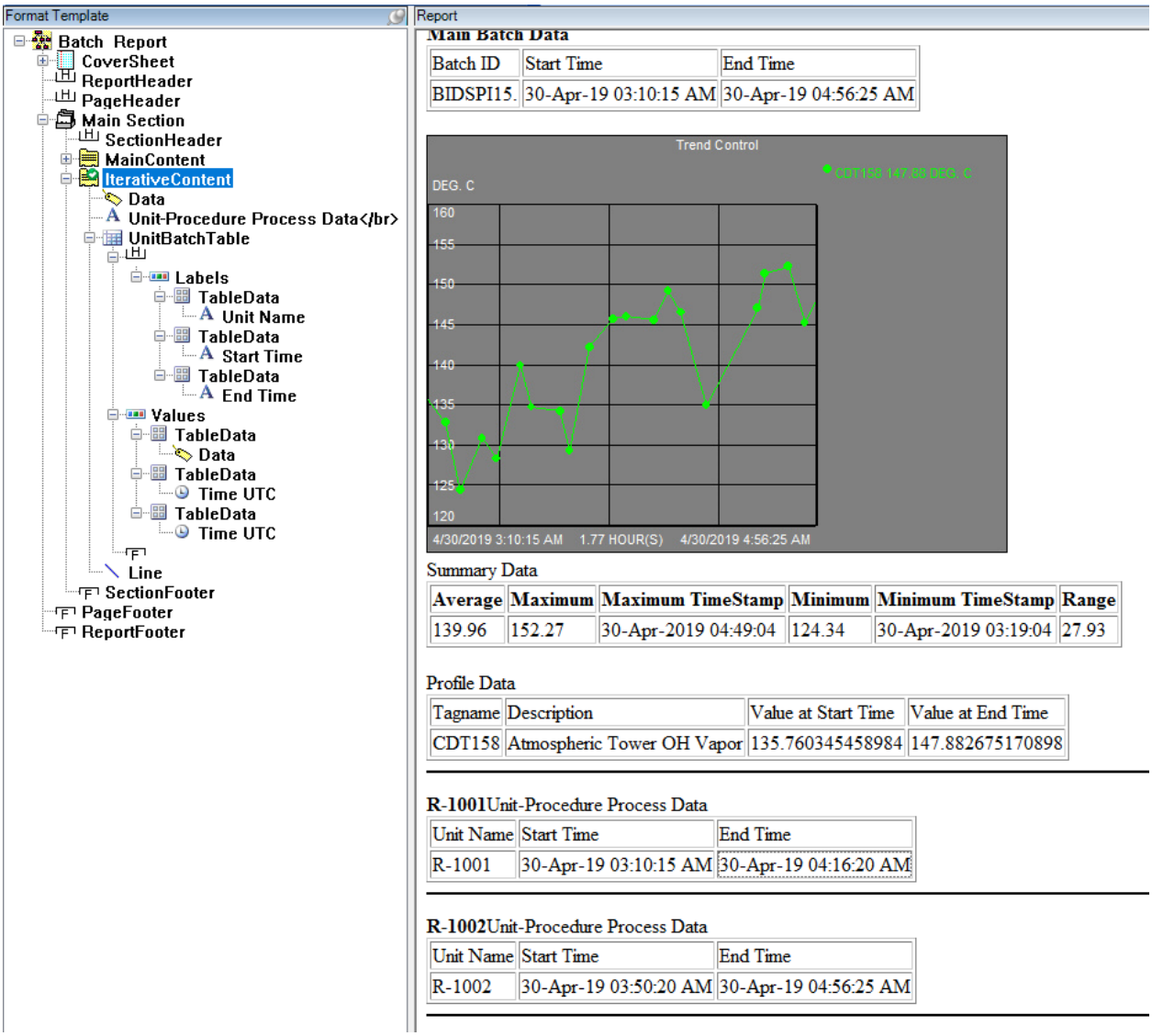
7. Journal Action Edit -> Action Results tab
8. Result Definition -> + -> Name: CompValues -> Type: *Compressed Values* -> drag Temperature (left) to Data Source (right) -> OK. Notice: [{ }] or {[ ]}



9. Save Report Template -> Test Report… or Refresh (F5)

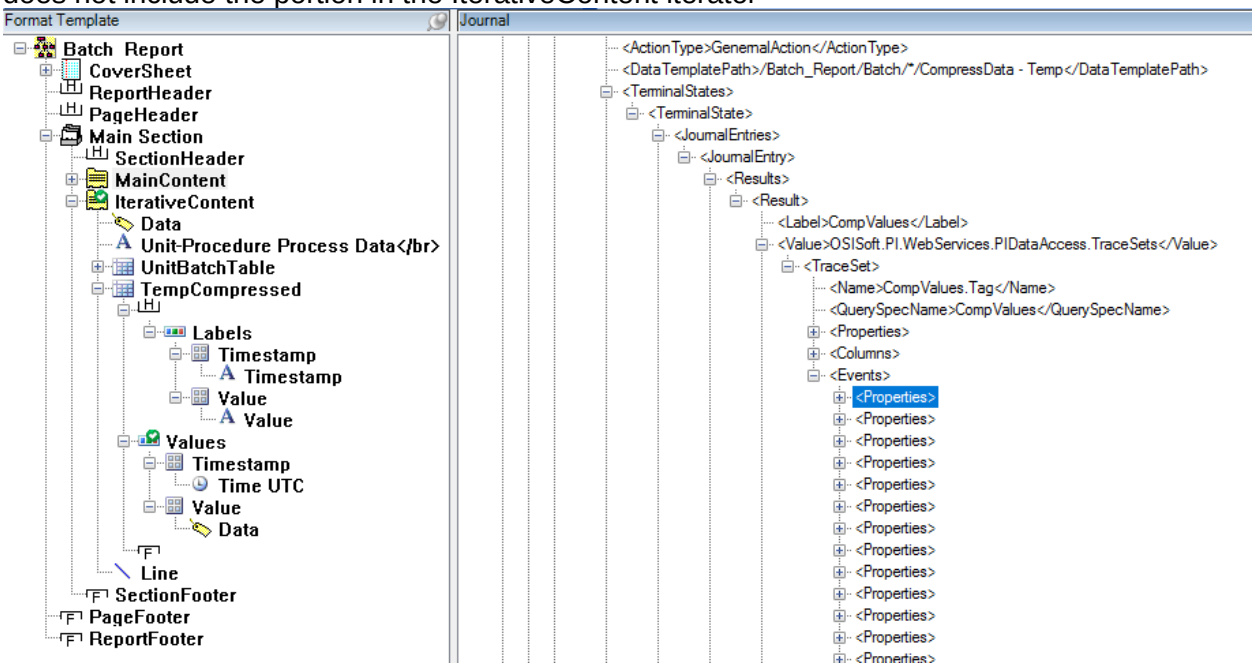
Modify iterator (from previous exercise), move the iterator from the UnitBatchTable's Values row to the IterativeContent entry

1. Switch to Format Template
2. Expand Main Section -> expand IterativeContent entry -> expand UnitBatchTable -> double clicks the Values row -> Expressions tab -> highlight and cut the Iterator: *BatchJournal/UnitBatchJournals/UnitBatchJournal* -> OK -> double clicks IterativeContent -> Expressions tab -> paste BatchJournal/UnitBatchJournals/UnitBatchJournal into Iterator -> OK
3. Save Report -> Refresh or F5

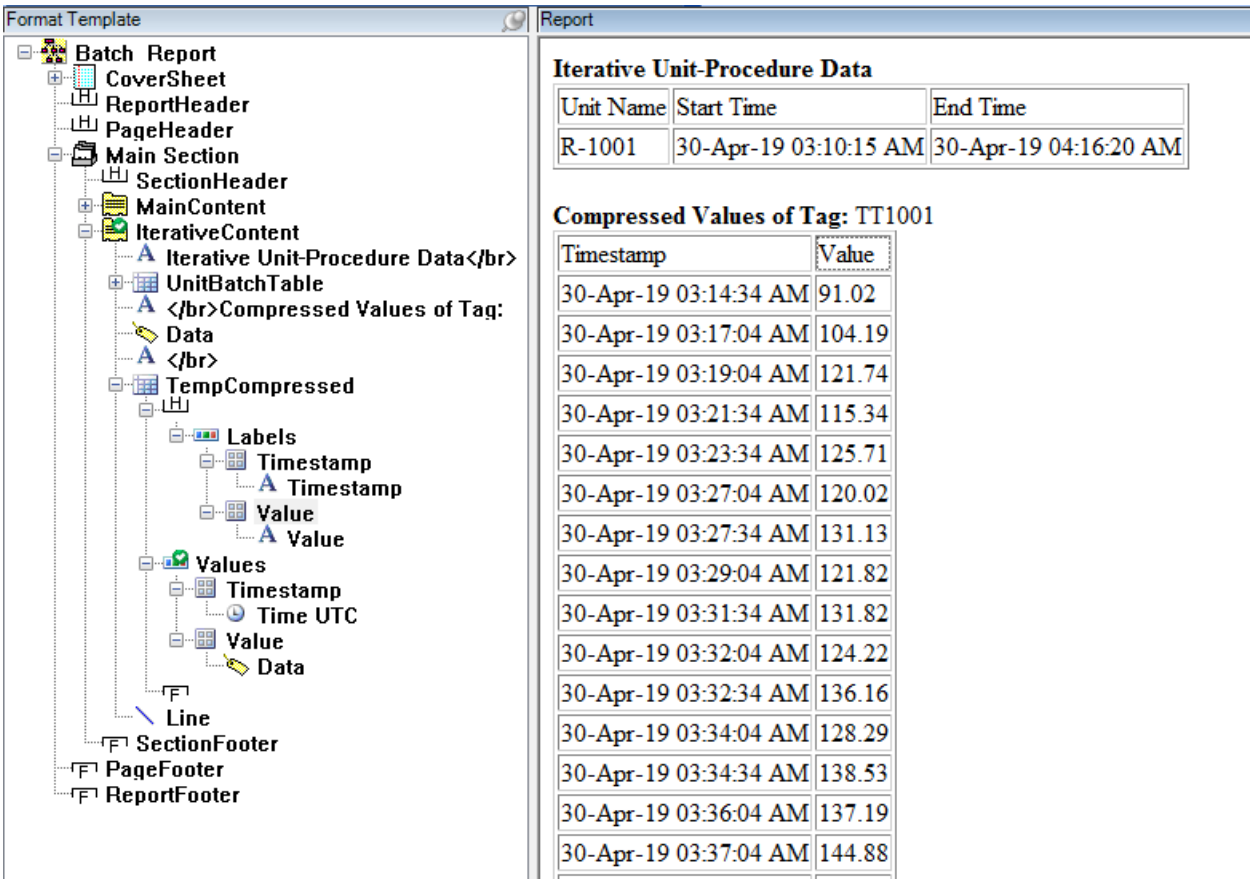

Format Template
Batch Report
CoverSheet
ReportHeader
PageHeader
Main Section
SectionHeader
MainContent
IterativeContent
Data
Unit-Procedure Process Data</br>
UnitBatchTable
Labels
TableData
Unit Name
TableData
Start Time
TableData
End Time
Values
TableData
Data
TableData
Time UTC
TableData
Time UTC
Line
SectionFooter
PageFooter
ReportFooter
Report
Main Batch Data
Batch ID
Start Time
End Time
BIDSPI15.
30-Apr-19 03:10:15 AM
30-Apr-19 04:56:25 AM
Trend Control
DEG. C
160
155
150
145
140
135
130
125
120
4/30/2019 3:10:15 AM
1.77 HOUR(S)
4/30/2019 4:56:25 AM
Summary Data
Average
Maximum
Maximum TimeStamp
Minimum
Minimum TimeStamp
Range
139.96
152.27
30-Apr-2019 04:49:04
124.34
30-Apr-2019 03:19:04
27.93
Profile Data
Tagname
Description
Value at Start Time
Value at End Time
CDT158
Atmospheric Tower OH Vapor
135.760345458984
147.882675170898
R-1001Unit-Procedure Process Data
Unit Name
Start Time
End Time
R-1001
30-Apr-19 03:10:15 AM
30-Apr-19 04:16:20 AM
R-1002Unit-Procedure Process Data
Unit Name
Start Time
End Time
R-1002
30-Apr-19 03:50:20 AM
30-Apr-19 04:56:25 AM

Adding a Table to display the compressed data

1. Select the Format Template tab
2. Expand Main Section
3. Select IterativeSection -> Insert -> Node Name: TempCompressedValues -> Rows: 1 -> Columns: 2 -> Border tab -> Frame: Border -> Border Width: 1 -> OK
4. Rename TableRow to Values -> expand Values row -> rename each TableData to Timestamp and Value
5. Select Values row -> copy -> select table header -> paste -> rename Values to Labels
6. Expand the table header Labels row -> select each TableData -> insert -> Text -> Text: same as column name
7. Select and double click the Values row -> Expressions tab -> browse and drag *JournalEvents/JournalEvent[Name="CompressData - Temp"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="CompValues"]/Value/TraceSet[@Name="CompValues.Tag"]/Events/Properties* to Iterator field. **Notice:** it does not include the portion in the IterativeContent iterator



8. Expand the Values row -> select the *Timestamp* TableData -> Insert -> *UTC Time* -> browse and drag *Property[@Descriptor="100"]* into the Source field -> OK -> select the Value cell -> Insert -> Data -> browse and drag *Property[@Descriptor="22000"]* to the Source field -> OK
9. Save Report -> Refresh or F5


Format Template
Batch Report
CoverSheet
ReportHeader
PageHeader
Main Section
SectionHeader
MainContent
IterativeContent
Iterative Unit-Procedure Data</br>
UnitBatchTable
</br>Compressed Values of Tag:
Data
</br>
TempCompressed
Labels
Timestamp
Timestamp
Value
Value
Values
Timestamp
Time UTC
Value
Data
Line
SectionFooter
PageFooter
ReportFooter
Report
Iterative Unit-Procedure Data
Unit Name
Start Time
End Time
R-1001
30-Apr-19 03:10:15 AM
30-Apr-19 04:16:20 AM
Compressed Values of Tag: TT1001
Timestamp
Value
30-Apr-19 03:14:34 AM
91.02
30-Apr-19 03:17:04 AM
104.19
30-Apr-19 03:19:04 AM
121.74
30-Apr-19 03:21:34 AM
115.34
30-Apr-19 03:23:34 AM
125.71
30-Apr-19 03:27:04 AM
120.02
30-Apr-19 03:27:34 AM
131.13
30-Apr-19 03:29:04 AM
121.82
30-Apr-19 03:31:34 AM
131.82
30-Apr-19 03:32:04 AM
124.22
30-Apr-19 03:32:34 AM
136.16
30-Apr-19 03:34:04 AM
128.29
30-Apr-19 03:34:34 AM
138.53
30-Apr-19 03:36:04 AM
137.19
30-Apr-19 03:37:04 AM
144.88

Lesson 6: Time Based Reporting

Time Report Templates allow users to configure Report Templates that can be generated for a specific time range or recurrence pattern. The Report Template is generated against an overall time range, but can be further divided into relative or fixed duration time spans. Users can now create detailed time reports with hierarchical time frames or interrelated time ranges.

Time Based Templates

Use the Time Report Details dialog box to create a report based on time criteria, rather than batches or Analysis Framework Cases.

You can define a fixed duration time template or a relative time template. A fixed duration time template creates a fixed time period anchored to either the parent start time or the parent end time.

Criteria for a time-based report is an interval specifying a report duration (and start date) and a pattern for when the report should be run. Providing this information serves two purposes. First, it allows the generator to assist the user in selecting a time range. Secondly, it provides a definition for what is considered an official report. For example, a weekly report defined to run for 5 days that was run by a user for only 3 days would not be considered an official report.

Recurrence

This drop-down list offers common time intervals for the recurrence pattern. Selections are:

- None
- Minutes
- Hours
- Days
- Weeks
- Months
- Years

The selected value changes the criteria selections in the group box below, allowing the user to select appropriate criteria for the recurrence of the report, as described below.

Official reports may be run only in the time range.

Reference Start

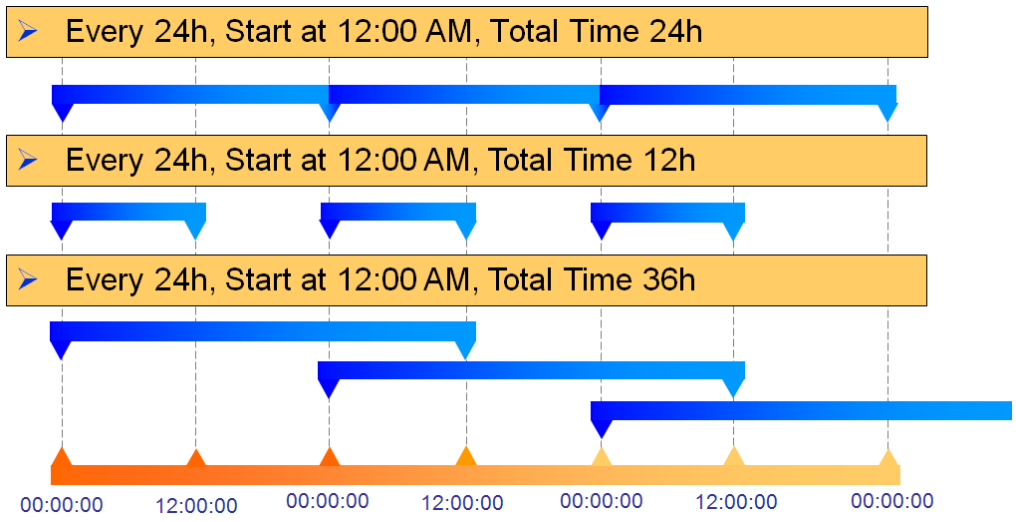
This drop-down list may also have a calendar control for selecting a date. This is a reference time (and possibly date) indicating when an official report can begin. Some recurrence selections require a date and others do not. In the event that a date is required, a calendar control is available. In other cases, only times are available for selection.

Report Duration

Determines how the duration is calculated for the report. The selection is a matter of convenience only. Either selection results in a fixed duration for the report.

When the Reference End option is selected, a drop-down list and/or calendar control are used to set the end time. The Reference End functions in the same way as the Referenced Start in that sometimes a date is required and sometimes it is not, depending on the Recurrence value. If the End time is less than or equal to the start time, it is assumed to occur on the following day (if dates are not entered). When the Total Time option is selected, quantity and time unit are entered for the duration (for example, 1 hour or 3 days). Fractional quantities are accepted for seconds, minutes or hours (as they are supported by the PI SDK) and the available units are based on the selection in the Recurrence drop-down list.

Applying Time Rules



The Time Journal

The Journal holds the results of the Business Rules applied to a context, in this case, a time range with sub-divisions. This information is stored under an XML structure and it is important to understand its organization in order to extract and display the critical data in our reports.

At the top level, we have the TimeBasedRootJournal. This level is the report boundaries as specified in the generator. You can get access to the Start Time, End Time and Duration along with other configuration information.

The second level of interest is the Journals which hold a collection of all the time sub-divisions produced by the Time Unit Template. These sub-divisions are found in each TimeBasedJournal.

Using Equipment in a Time Report

You can take advantage of an equipment context in a time-based report by using a Module Template. Inserting a Module Template within a time based report will allow you to use the aliases and properties within that unit context in the report.

Exercise 7: Time Report Templates

Objectives

Learn how to add interval to the Time-based Template

Problem Description

Create a Business Day Report; a 24 hours business day report that span from 12:00 a.m. to 12:00 a.m. the next day. The daily report is subdivided into 2 intervals from 12:00 a.m. to 12:00 p.m. and from 12:00 p.m. to 12:00 a.m. of the next day.

Approach

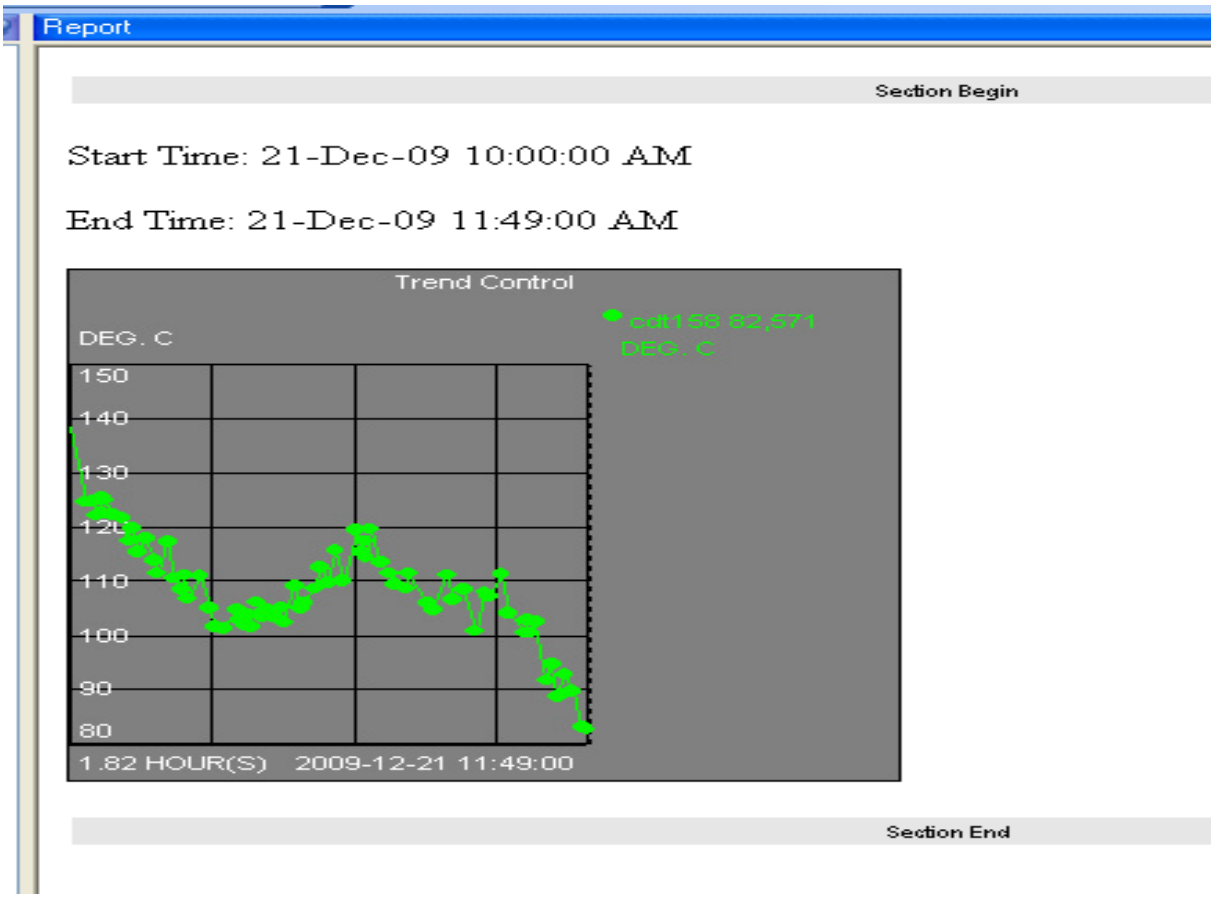
Create a new Time Report Template called BusinessDayReport.

Configure the recurrence pattern to generate the daily report only on business days.

Add/insert a Time-Based Template to subdivide the day into 2 periods of equal duration.

Display a trend of the data for CDT158 in the report. Also display the start time and the end time.

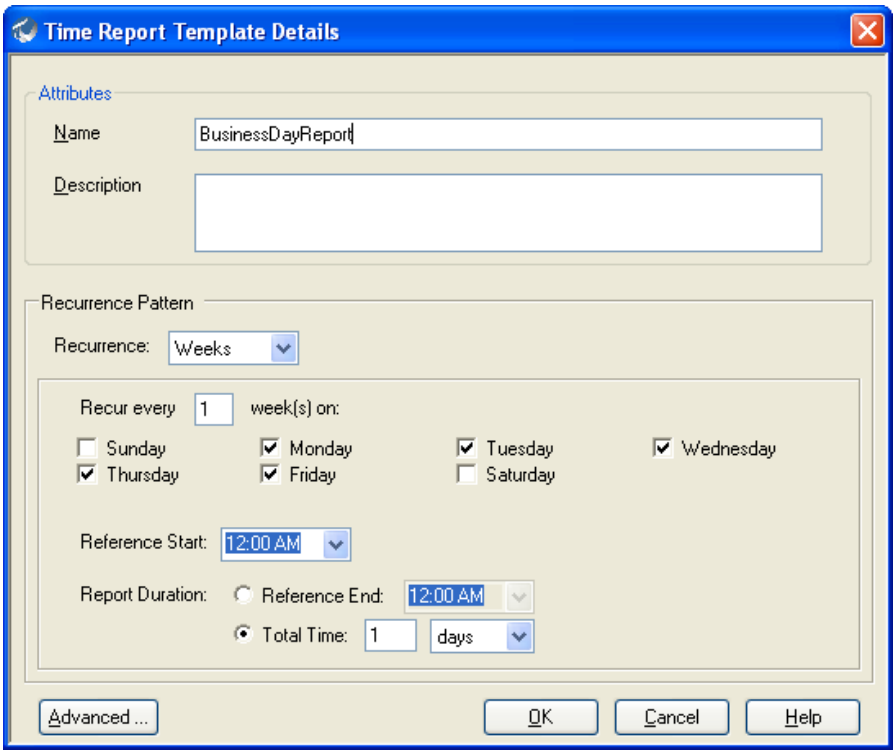
An example is shown below.


Report
Section Begin
Start Time: 21-Dec-09 10:00:00 AM
End Time: 21-Dec-09 11:49:00 AM
Trend Control
DEG. C
cdt158 82,571
DEG. C
150
140
130
120
110
100
90
80
1.82 HOUR(S) 2009-12-21 11:49:00
Section End

Solution 7: Time Report Template Creation

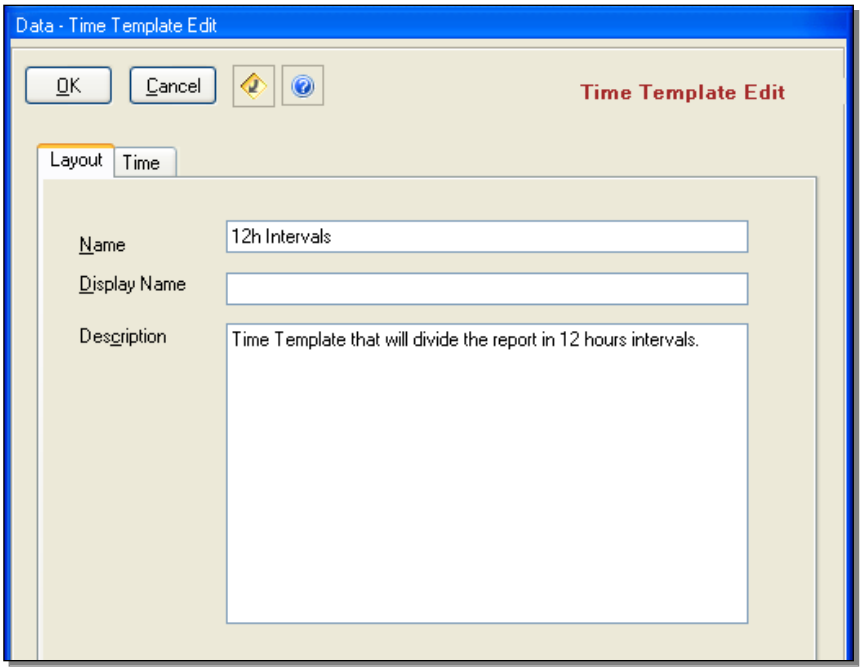
Create a new Time Report Template

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Create a new template File -> New -> Time Report Template… -> Name: BusinessDayReport
3. In the Recurrence Pattern section, Recurrence -> Weeks -> Recur every 1 week(s) on: check Monday, Tuesday, Wednesday, Thursday and Friday -> Reference Start: 12:00 AM -> and Total Time: 1 days -> OK

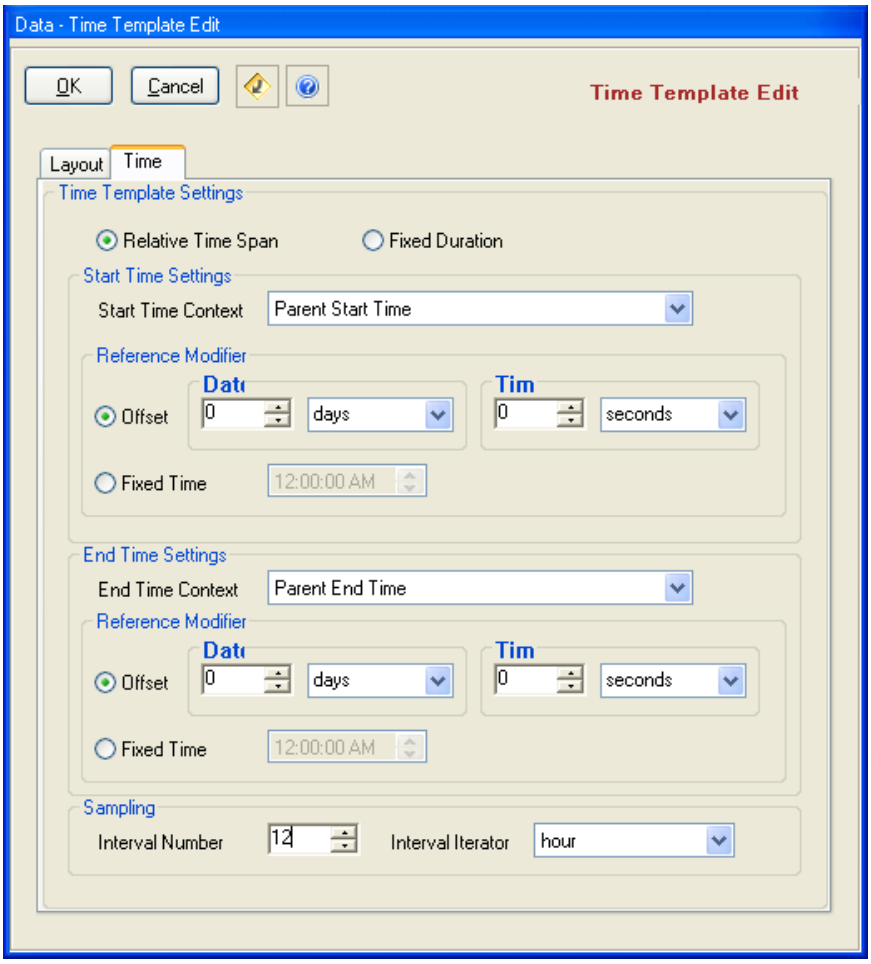


Create sub-division time intervals using the Time-Based Template

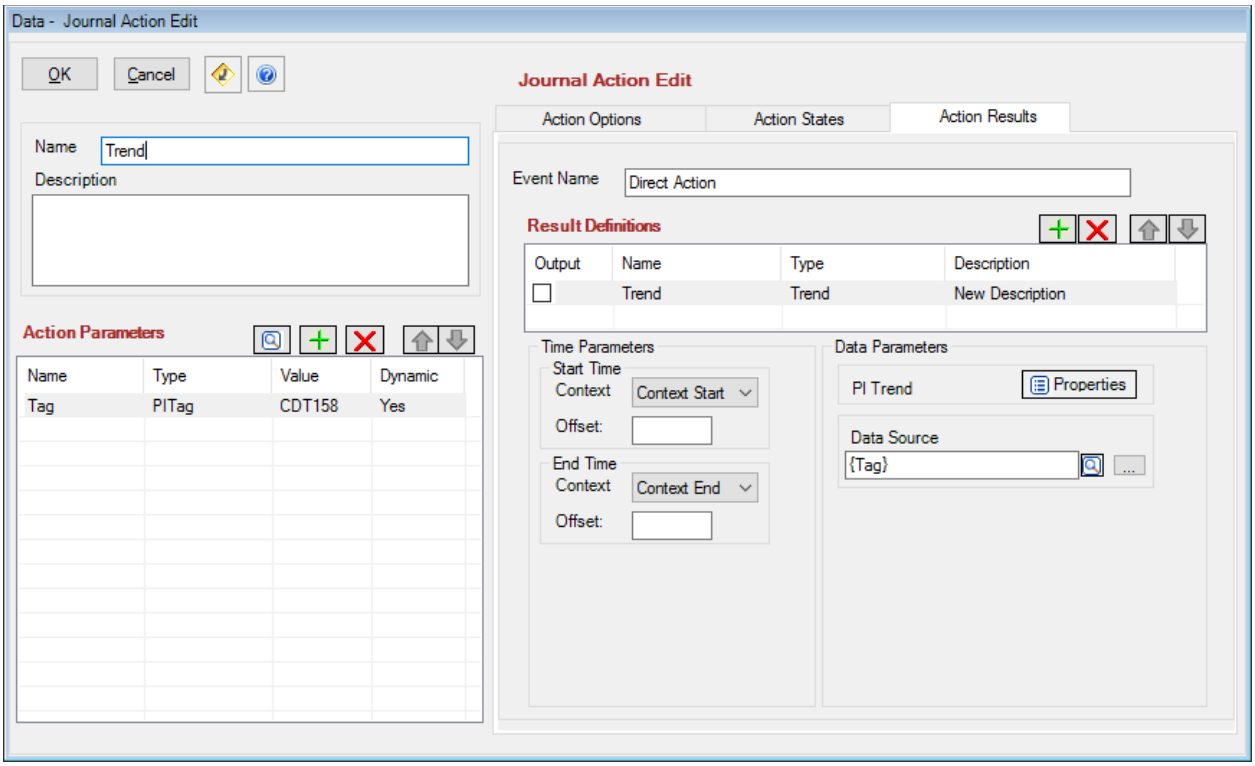
1. In the Data Template tab
2. Select the “Time” Template -> Insert -> Time Based Template -> Name: 12h Intervals



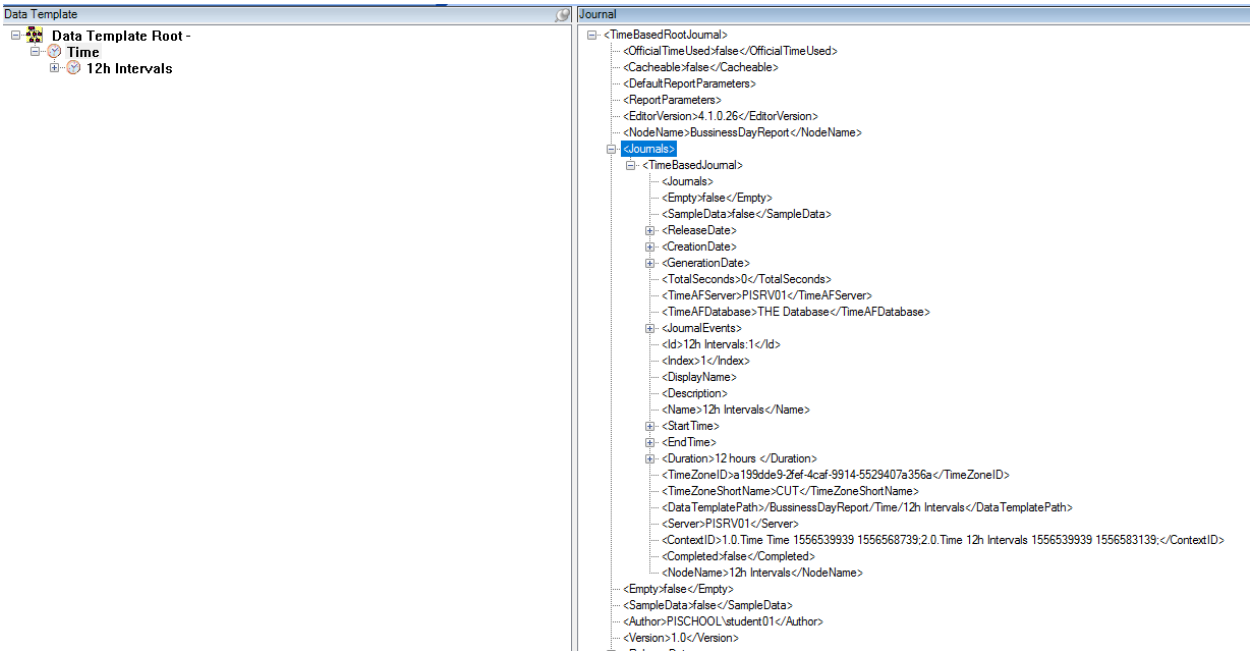
3. Switch to the Time tab -> in the Sampling section -> Interval Number: 12 hour -> OK



4. Select the “12h Intervals” Time Based Template
5. Insert -> Journal Action -> Name: Trend
6. Action Parameter: + -> Name: Tag -> Type: PITag -> browse/find CDT158 -> OK
7. Journal Action Edit: Action Results tab
8. Result Definitions: + -> Name: Trend -> Type: Trend -> drag “Tag” (left) to Data Source field (right -> OK
9. Save Report Template -> View -> Test Report… (or F2) -> Asset Server: PISRV01 -> Asset Database: THE Database -> Start Time:

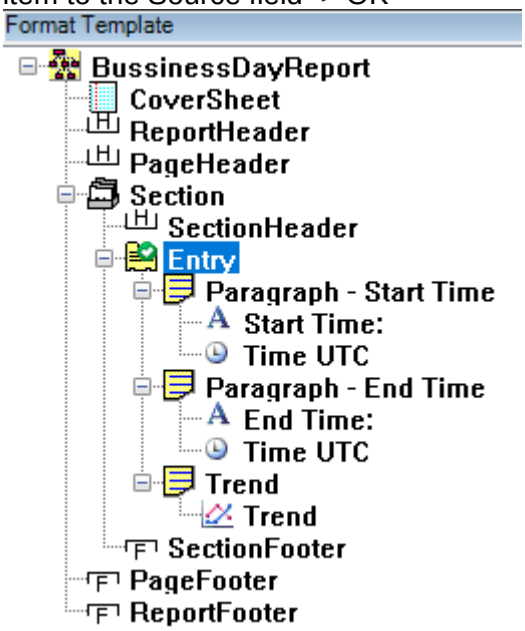


10. Test the report, View -> Test Report… or F2 -> Data Server: PISRV01 -> Asset Server: PISRV01 -> Asset Database: THE Database -> Start Time: *-8h -> End Time: * -> OK
11. Notice the Journal tab



12. Switch to the Format Template tab
13. Select root

14. Insert Section -> Expand “Section” -> Select “Entry” -> Insert -> Paragraph -> Node Name: Paragraph – Start Time -> OK
15. Select “Entry” -> Insert -> Paragraph -> Node Name: Paragraph – End Time -> OK
16. Select “Entry” -> Insert -> Paragraph -> Node Name: Paragraph – Trend -> OK
17. Select the “Paragraph – Start Time” -> Insert -> Text -> Text: Start Time: -> Insert UTC Time -> browse and drag *StartTime/UTCSeconds* item to the Source field -> OK
18. Setup iteration, double click “Entry” -> Expressions tab -> browse and drag *TimeBasedRootJournal/Journals/TimeBasedJournal* to Iterator field
19. Select the “Paragraph – Start Time” -> Insert -> Text -> Text: Start Time: -> Insert UTC Time -> browse and drag *EndTime/UTCSeconds* item to the Source field -> OK
20. Select the “Paragraph – Trend” -> Insert -> Text -> Text: Start Time: -> Insert UTC Time -> browse and drag *JournalEvents/JournalEvent[Name="Trend"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Trend"]/Value* item to the Source field -> OK

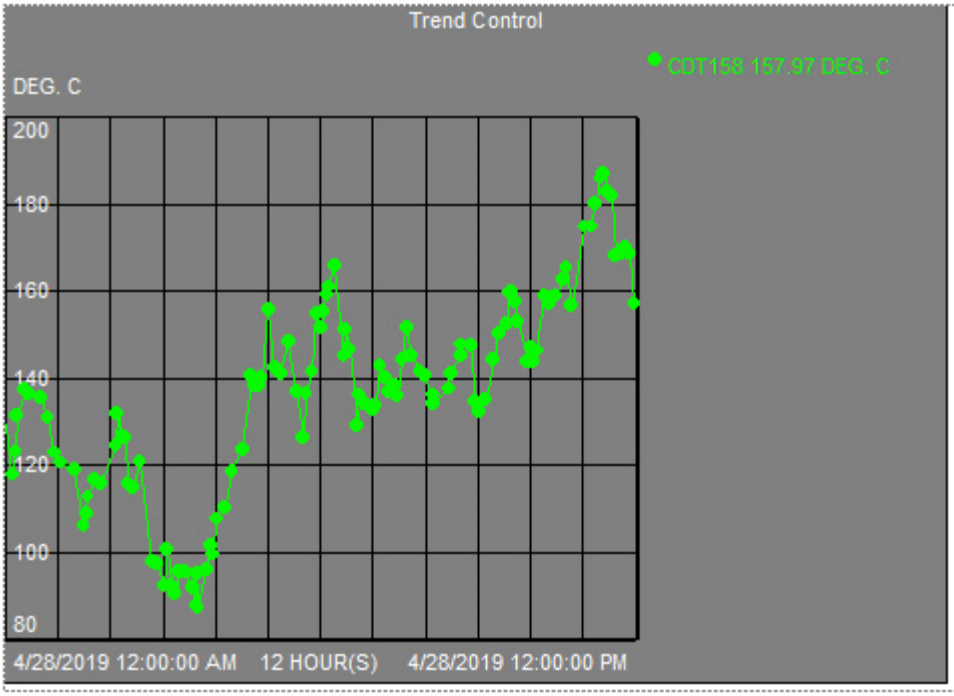


21. Save Report Template -> Refresh or F5

Report

Start Time: 28-Apr-19 12:00:00 AM

End Time: 28-Apr-19 12:00:00 PM



Start Time: 28-Apr-19 12:00:00 PM

End Time: 29-Apr-19 12:00:00 AM

Lesson 7: Condition Based Reporting

We will see how to get more value out of your reports by using condition based reports.

Reporting Without Conditions:

- Large volume of documents
- High risk of missing the key information
- Overlook small details
- Spend more time browsing through the report

Condition Based Reporting:

- Saves Time
- Goes right to the point
- Provides more details when required

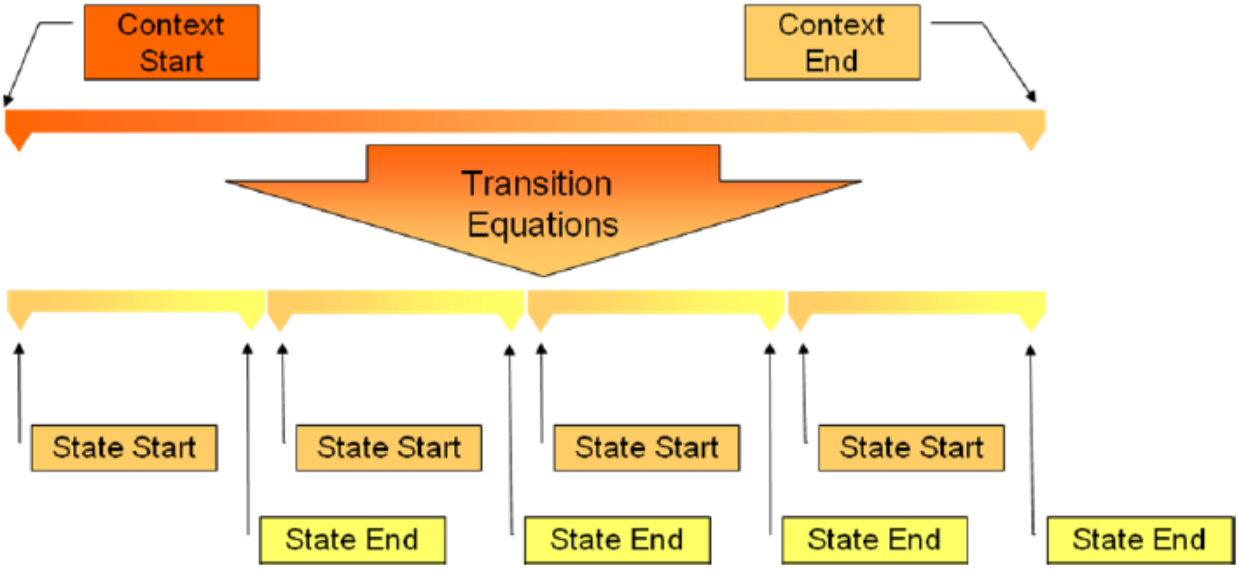
Building Condition Based templates offer two approaches:

In the **Data Template → Action States**
(Specify which data to extract and display it)

In the **Format Template → Criteria**
(Extract all the data and select what will be displayed)

Using Action States

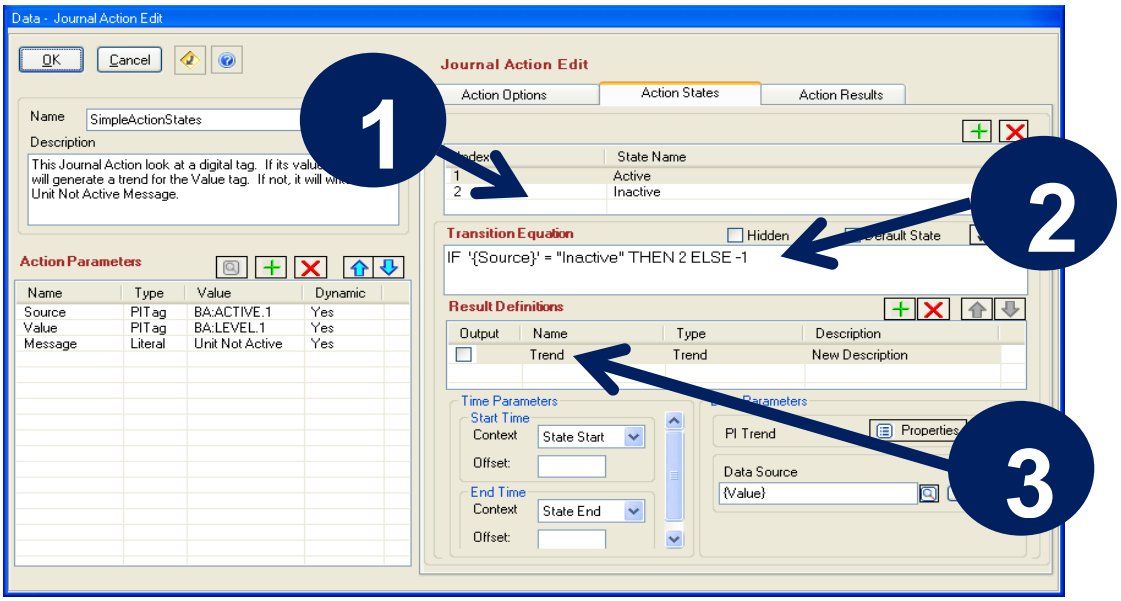
The context element defines the overall time range for a Journal Action. Action States are the individual stages run within a context element. Each Action State has a transition equation that is run when it is in that state. The transition equation returns the next state number.



The time intervals for execution of these equations are defined by the Interval Action Option. At each state transition, the Result Definitions for that state are generated.

There are three general steps involved in defining Actions States:

1. Define the number of States and names for each
2. Define the logic behind each state in the form: **IF {expression1} Then {statenumber1} Else If {expression2} Then {statenumber2} Else {statenumber3}** the format follows the syntax rules of performance equations
3. Define a result (what happens) for each transition condition.



This is extremely flexible when Action Parameters are used (aliases or tags).

When are the Transitions Evaluated?

The Interval property located in **Action Options** defines the logic for retrieving the Interval timestamps for the transition equations.

Interval - There are three possibilities:

- **(-1):** The transition equations will be executed using the recorded value call of the PI SDK. Hence, the equations will be executed at specific times that have data in the PI Archive.
- **(Integer Number):** The transition equations will be executed at evenly spaced intervals between the period start and end defined by this time interval in seconds.
- **(Tag Name):** The transition equations will be executed at the recorded values of this tag. This tag does not have to be included in the transition equations defined for individual states.

Optional Output of Actions

The Output property defines the logic for inclusion in the report. If there is a problem executing this Journal Action and if this property is set to **Optional**, the report will log a message to the PI Server and continue executing the report. If this property is set to "**Required"**, then any error associated with this Journal Action would stop the generation of the report and the user would be notified that there was a problem executing the report.

Filter or Suppress Transition Equations

Suppression settings (**Suppress Set**) for the Action define logic to suppress the execution of the Journal Action until certain conditions are met.

Suppress Monitoring Expression – Specifies a PI PE Expression that should be evaluated at all time-intervals during the action to see if those time intervals should be included.

Start Percent of Time – Defines whether the integer value located in the SuppressfromStart property is in seconds or percent of period time.

Suppress from Start – Integer value signifying the seconds or percent of time to suppress the action logic from the start of the period.

End Percent of Time - Defines whether the integer value located in the SuppressfromEnd property is in seconds or percent of period time.

Suppress from End - Integer value signifying the seconds or percent of time to suppress the action logic from the end of the period.

The Other Action Options

You can suppress based on a start time or end time deadband (in seconds) or you can generate an error if the result cannot be computed (Output Type = “Required”).

Exercise 8: Using the State Machine

Objectives

Learn how to configure custom States and to apply Result Definitions to each State accordingly.

Problem Description

Monitor PI Tag “Ba:Level.1” using a custom 3-State State Machine as defined in the following table. Evaluated the Transition Equation every 30 seconds.

State	Low Limit	High Limit	Message	Action
High	> 30		High Alarm	Display the Message Get a trend of the Ba:Level.1 tag
In Range	> 15	<= 30	In Range	Display the Message
Low		<=15	Low Alarm	Display the Message

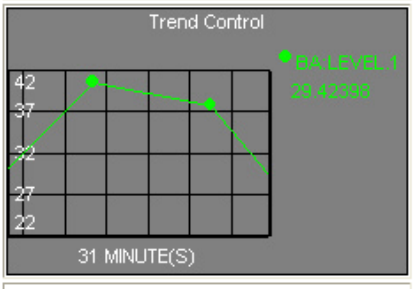
Approach

Create a Journal Action using the Action States options

Create a table to show the results.

Report Preview

My Alarms

State	Start Time	End Time	Trend
In Range	21-Feb-06 01:34:20 PM	21-Feb-06 01:53:20 PM	---
High Alarm	21-Feb-06 01:53:20 PM	21-Feb-06 02:24:20 PM	
In Range	21-Feb-06 02:24:20 PM	21-Feb-06 02:28:50 PM	---
Low Alarm	21-Feb-06 02:28:50 PM	21-Feb-06 02:54:50 PM	---
In Range	21-Feb-06 02:54:50 PM	21-Feb-06 02:59:00 PM	---

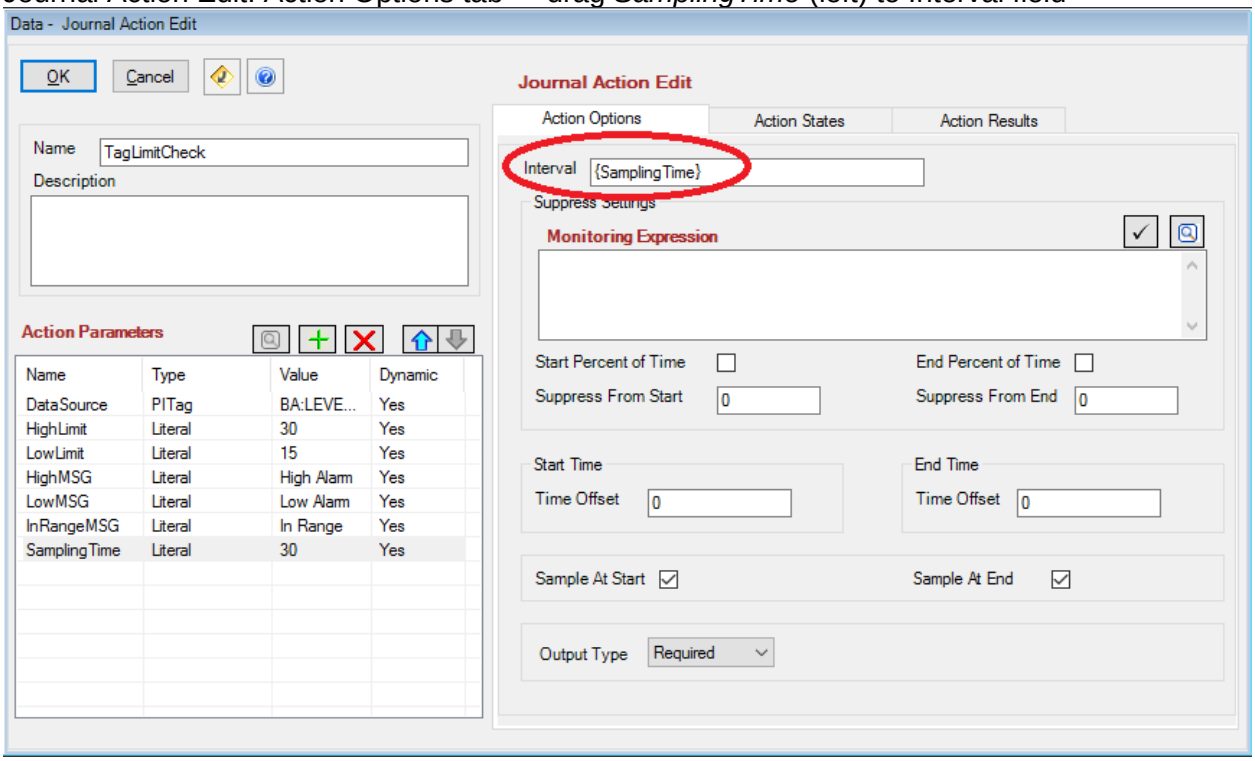
Solution 8: Using the State Machine

Create the Action State

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Open the *Batch_Report* template from pervious exercises
3. In the Data Template, select the "*Batch*" -> Insert-> Journal Action -> Name: *TagLimitCheck*.
4. Action Parameters: **+** to add the following Action Parameters

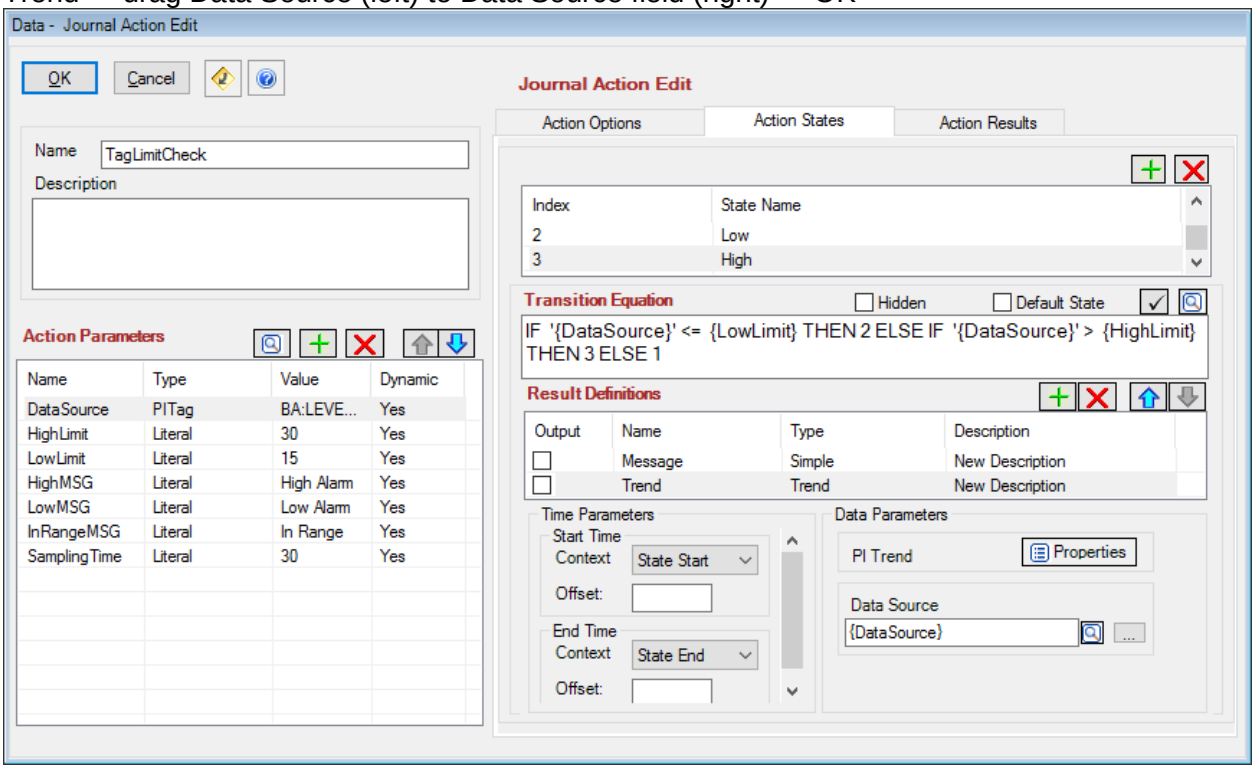
Name	Type	Value
DataSource	PI Tag	BA:Level.1
HighLimit	Literal	30
LowLimit	Literal	15
HighMSG	Literal	High Alarm
LowMSG	Literal	Low Alarm
InRangeMSG	Literal	In Range
Sampling Time	Literal	30

5. Journal Action Edit: Action Options tab -> drag *SamplingTime* (left) to Interval field



6. Journal Action Edit: switch to the Action States tab -> **+** -> State Name: InRange (Index: 1) -> **+** -> State Name: Low (Index: 2) -> **+** -> State Name: High (Index: 3)

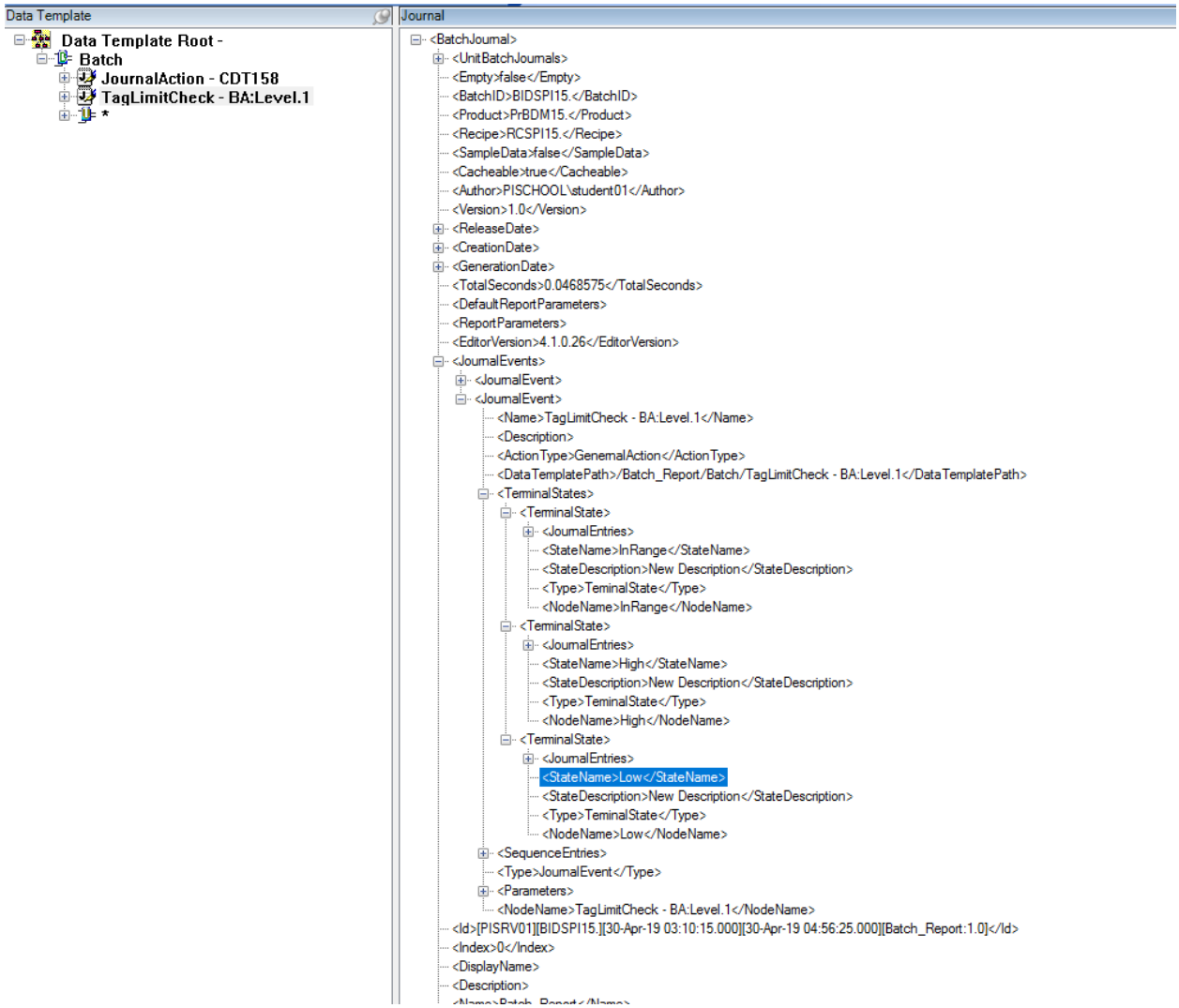
7. Select the “InRange” (Index: 1) State -> check the “Default State” -> click the *Transition Equation* space -> type in *IF '{DataSource}' <= {LowLimit} THEN 2 ELSE IF '{DataSource}' > {HighLimit} THEN 3 ELSE 1* -> copy the entire equation and paste into the *Transition Equation* space of the State Name: Low and High
8. Select the “InRange” State -> Result Definitions section -> green **+** -> Name: Message -> Type: Simple -> drag InRangeMSG (left) to Value field (right)
9. Select the “Low” State -> Result Definitions section -> green **+** -> Name: Message -> Type: Simple -> drag LowMSG (left) to Value field (right)
10. Select the “High” State -> Result Definitions section -> green **+** -> Name: Message -> Type: Simple -> drag HighMSG (left) to Value field (right) -> green **+** -> Name: Trend -> Type: Trend -> drag Data Source (left) to Data Source field (right) -> OK



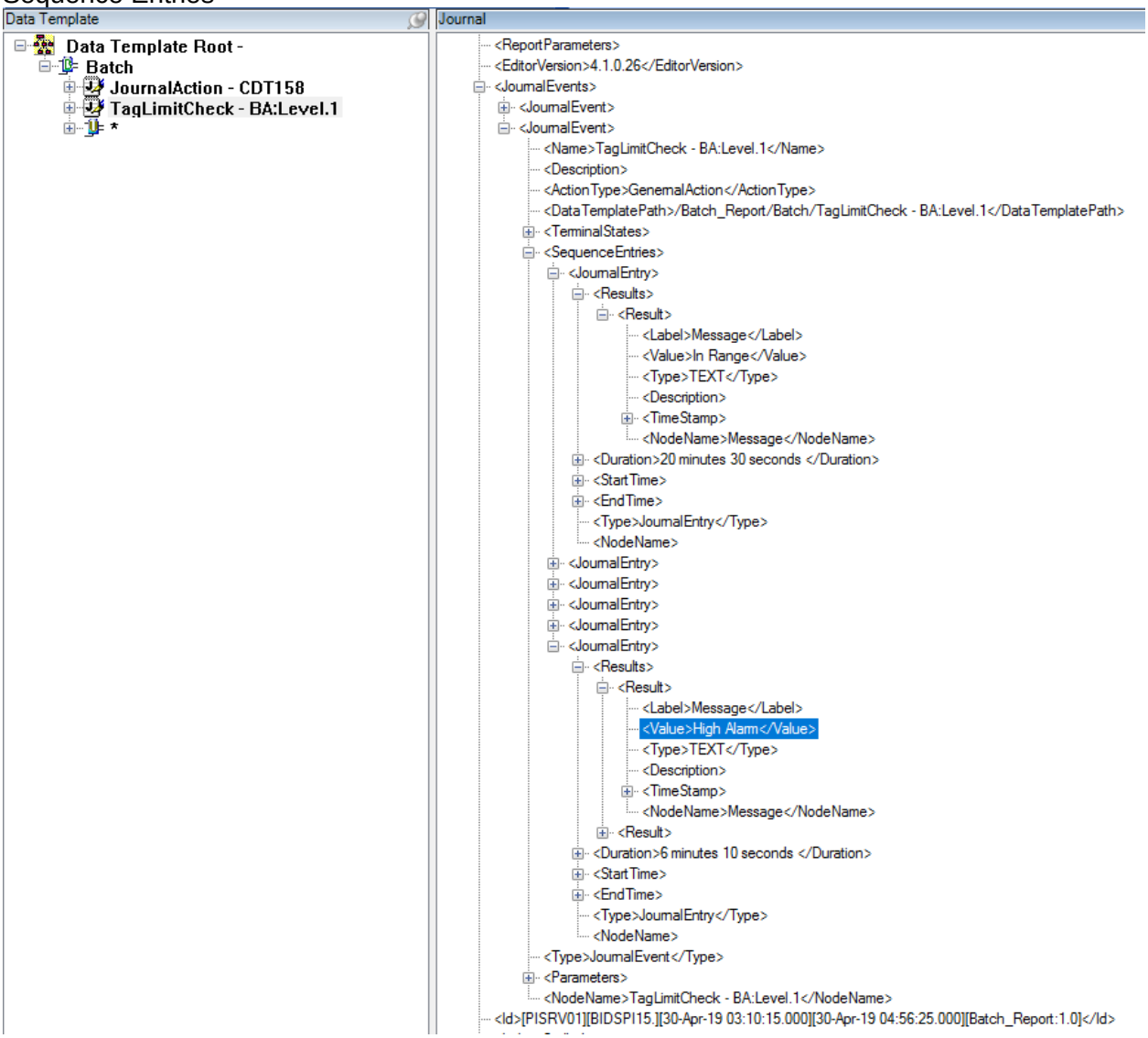
11. Save Report Template -> View -> Test Report -> Use Event Frames (or F3) -> Server: PISRV01 -> Database: THE Database… -> Search -> select a completed Procedure EF -> OK

Journal Observation

1. Select the Journal tab
2. Browse the Journal up to the result of the TagLimitCheck Journal Action, you will notice the TerminalStates and the SequenceEntries nodes. Both nodes contain the same information although, it is organized differently. The TerminalStates should have three entries (In Range, Low and High). All state results will be gathered under one of them. The SequenceEntries will show the same information but in sequential order. Since we need a table listing the different states, we will work with the SequenceEntries node
3. Terminal States

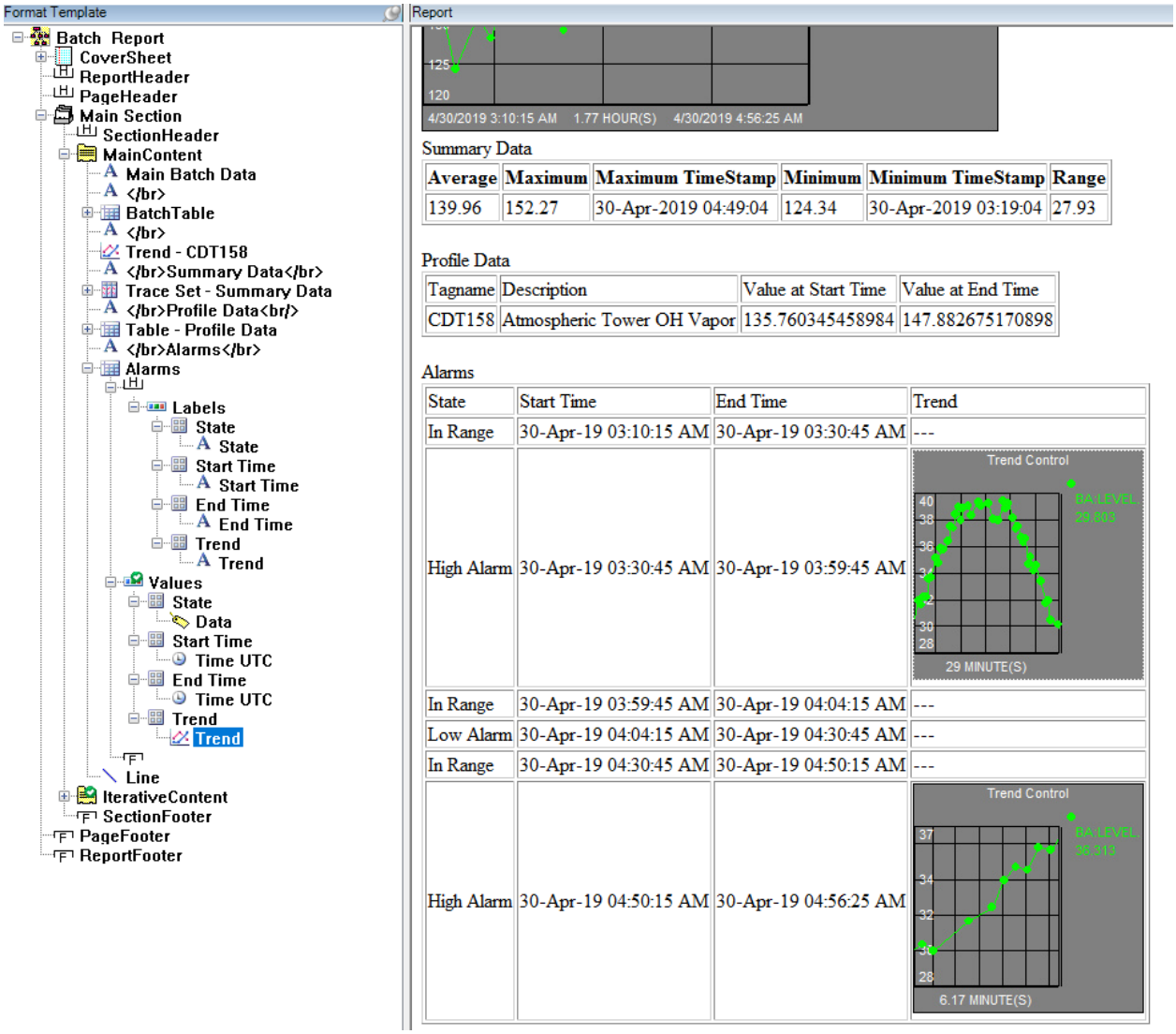


4. Sequence Entries



Format

1. Switch to the Format Template tab
2. Expand the “Main Section” -> select the “MainContent” entry -> Insert -> Table -> Node Name: Alarms -> Rows: 1 -> Columns: 4 -> Border tab -> Frame: Border -> Border Width: 1 -> OK
3. Expand the Alarms table -> rename the TableRow to Values
4. Expand the Values row -> rename each TableData to State, Start Time, End Time, and Trend
5. Select the Values row -> copy -> select the table header -> paste -> rename Values to Labels
6. Expand the Labels row -> select each TableData -> Insert -> Text -> Text: *same as columns name*
7. To setup Iterator for the SequenceEntries JournalEntry, double click the “Values” row -> Expressions tab -> browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="TagLimitCheck"]/SequenceEntries/JournalEntry* to Iterator field
8. Select the “State” TableData (cell) -> Insert -> Data -> browse and drag *Results/Result[Label="Message"]/Value* to the Source field
9. Select the “Start Time” TableData (cell) -> Insert -> UTC Time -> browse and drag *StartTime/UTCSeconds* to the Source field
10. Select the “End Time” TableData (cell) -> Insert -> UTC Time -> browse and drag *EndTime/UTCSeconds* to the Source field
11. Select the “Trend” TableData (cell) -> Insert -> Trend -> browse and drag *Results/Result[Label="Trend"]/Value* to the Source field
12. Save Report Template -> Refresh or F5. Notice the un-desired “---” in place of a Trend (in the Trend column) when State is In Range or Low


Format Template
Batch Report
CoverSheet
ReportHeader
PageHeader
Main Section
SectionHeader
MainContent
Main Batch Data
</br>
BatchTable
</br>
Trend - CDT158
</br>Summary Data</br>
Trace Set - Summary Data
</br>Profile Data
Table - Profile Data
</br>Alarms</br>
Alarms
Labels
State
State
Start Time
Start Time
End Time
End Time
Trend
Trend
Values
State
Data
Start Time
Time UTC
End Time
Time UTC
Trend
Trend
Line
IterativeContent
SectionFooter
PageFooter
ReportFooter
Report
125
120
4/30/2019 3:10:15 AM 1.77 HOUR(S) 4/30/2019 4:56:25 AM
Summary Data
Average
Maximum
Maximum TimeStamp
Minimum
Minimum TimeStamp
Range
139.96
152.27
30-Apr-2019 04:49:04
124.34
30-Apr-2019 03:19:04
27.93
Profile Data
Tagname
Description
Value at Start Time
Value at End Time
CDT158
Atmospheric Tower OH Vapor
135.760345458984
147.882675170898
Alarms
State
Start Time
End Time
Trend
In Range
30-Apr-19 03:10:15 AM
30-Apr-19 03:30:45 AM

High Alarm
30-Apr-19 03:30:45 AM
30-Apr-19 03:59:45 AM
Trend Control
29 MINUTE(S)
In Range
30-Apr-19 03:59:45 AM
30-Apr-19 04:04:15 AM

Low Alarm
30-Apr-19 04:04:15 AM
30-Apr-19 04:30:45 AM

In Range
30-Apr-19 04:30:45 AM
30-Apr-19 04:50:15 AM

High Alarm
30-Apr-19 04:50:15 AM
30-Apr-19 04:56:25 AM
Trend Control
6.17 MINUTE(S)

Using Criteria

In any report, there may be a need to apply specific formatting based on the value of the data within the report. This is usually referred to as “conditional formatting.” For example, if you are creating a manufacturing exception report, then upon an exception, you may want to turn the values red or differentiate them from others using a different font. Condition expressions within format elements allow you to achieve this conditional formatting.

The condition expression is also based on the same syntax as the Source Expression. You specify a Source Expression, which evaluates the existence of an element or group of elements within Journal, and then place a Boolean operator around it. Note that the Boolean condition expression must be lower case. If the Boolean operator returns a True value, then the format element is included in the final rendered report. If the Boolean operator returns a “False” value, then the format element is not included in the final report.

Using the Condition Field

This is implemented in the Condition field of a format item. Only one condition can exist in an item.

For example, take a simple Journal Action that by its definition can lead to two different states: Active or Inactive. Whenever the source tag is set to Active, the user wishes to display a trend. Also, whenever the source tag is set to Inactive, the users wants to display a simple message saying that the unit was not active during this time period.

In this case, ***we need to add two*** formatting objects, one for the message and one for the trend. In the Data object for example:

```
boolean(Results/Result[Label="Message"])
```

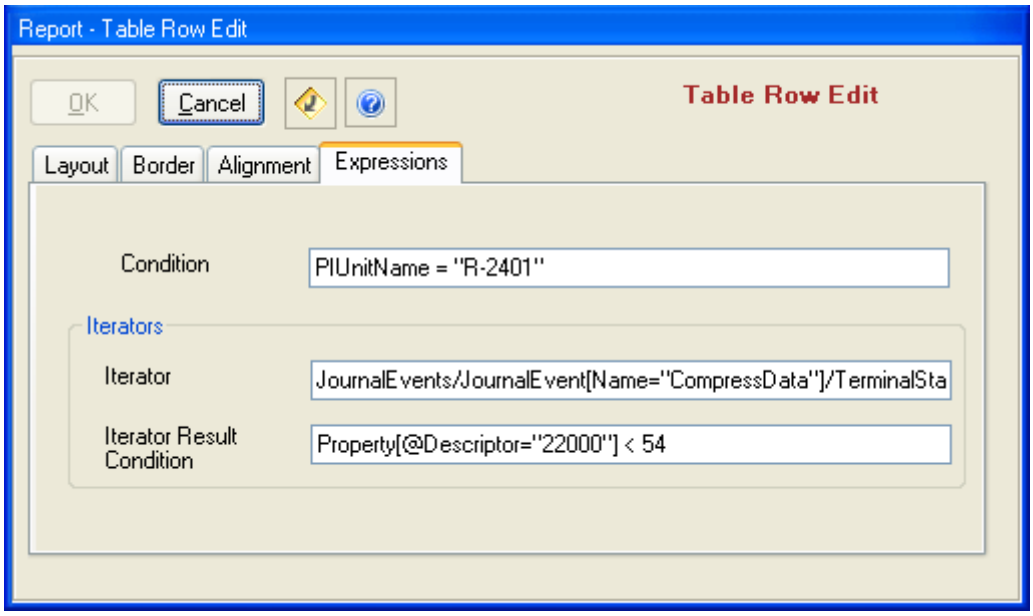
In the Trend object for example:

```
boolean(Results/Result[Label="Trend"])
```

These are representative only – you will have to figure out the exact phrasing for your own application.

Iterator Result Conditions

In some cases, we want to decide whether to show a value based upon the condition of an iteration.



This example is taken from a display of compressed data. As you will remember, the Compressed Data was displayed using a table with two rows. The first one held the column headers while the second one held the data. We used an Iterator on the second row to show all the Compressed Data, not only the first one.

The first condition (PIUnitName = “R-2401”) will decide if we run through the iteration or not. In this case, the iteration will not be performed if the data is coming from the unit R-2402.

The second condition, Iterator Result Condition (Property[@Descriptor=“220000”] < 54), will be used to decide which row will be shown. For example, if the first three values are greater than 54, the iteration will add one row for each value. If the fourth value is 32, then the iteration process will continue without adding a row. If the fifth value is 58, then the fourth row will be added.

XPath Functions for Criteria

boolean **starts-with**(string, string)	The **starts-with** function returns true if the first argument string starts with the second argument string, and otherwise returns false.
number **count**(node-set)	The **count** function returns the number of nodes in the argument node-set.
boolean **contains**(string, string)	The **contains** function returns true if the first argument string contains the second argument string, and otherwise returns false.
string **substring-before**(string, string)	The **substring-before** function returns the substring of the first argument string that precedes the first occurrence of the second argument string in the first argument string, or the empty string if the first argument string does not contain the second argument string. For example, substring-before("1999/04/01","/") returns 1999.
string **substring-after**(string, string)	The **substring-after** function returns the substring of the first argument string that follows the first occurrence of the second argument string in the first argument string, or the empty string if the first argument string does not contain the second argument string. For example, substring-after("1999/04/01","/") returns 04/01, and substring-after("1999/04/01","19") returns 99/04/01.
string **substring**(string, number, number?)	The **substring** function returns the substring of the first argument starting at the position specified in the second argument with length specified in the third argument. For example, substring("12345",2,3) returns "234". If the third argument is not specified, it returns the substring starting at the position specified in the second argument and continuing to the end of the string. For example, substring("12345",2) returns "2345".

Exercise 9: Using Conditional Formatting

Objectives

Use conditional formatting to decide which element of the report to display.

Problem Description

In the Batch_Report from previous exercises, apply the conditional formatting options to change the appearance of the report by:

- Replacing the --- symbol (non-existence journal element) with “NA” for the In Range and Low Alarm.
- Removing the In Range State from the list.

Approach

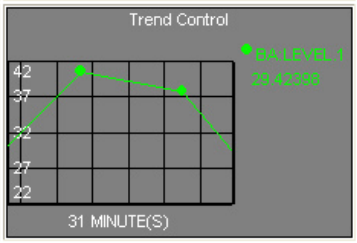
Add “NA” Text object to the Trend column of the table.

Condition to show the only the Trend object when the trend object does exist, else show only the “NA” text object.

Use the Iterator Result Condition field to skip (not display) any In Range message

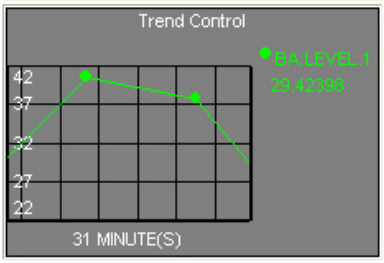
Original

My Alarms

State	Start Time	End Time	Trend
In Range	21-Feb-06 01:34:20 PM	21-Feb-06 01:53:20 PM	---
High Alarm	21-Feb-06 01:53:20 PM	21-Feb-06 02:24:20 PM	
In Range	21-Feb-06 02:24:20 PM	21-Feb-06 02:28:50 PM	---
Low Alarm	21-Feb-06 02:28:50 PM	21-Feb-06 02:54:50 PM	---
In Range	21-Feb-06 02:54:50 PM	21-Feb-06 02:59:00 PM	---

Transformed into

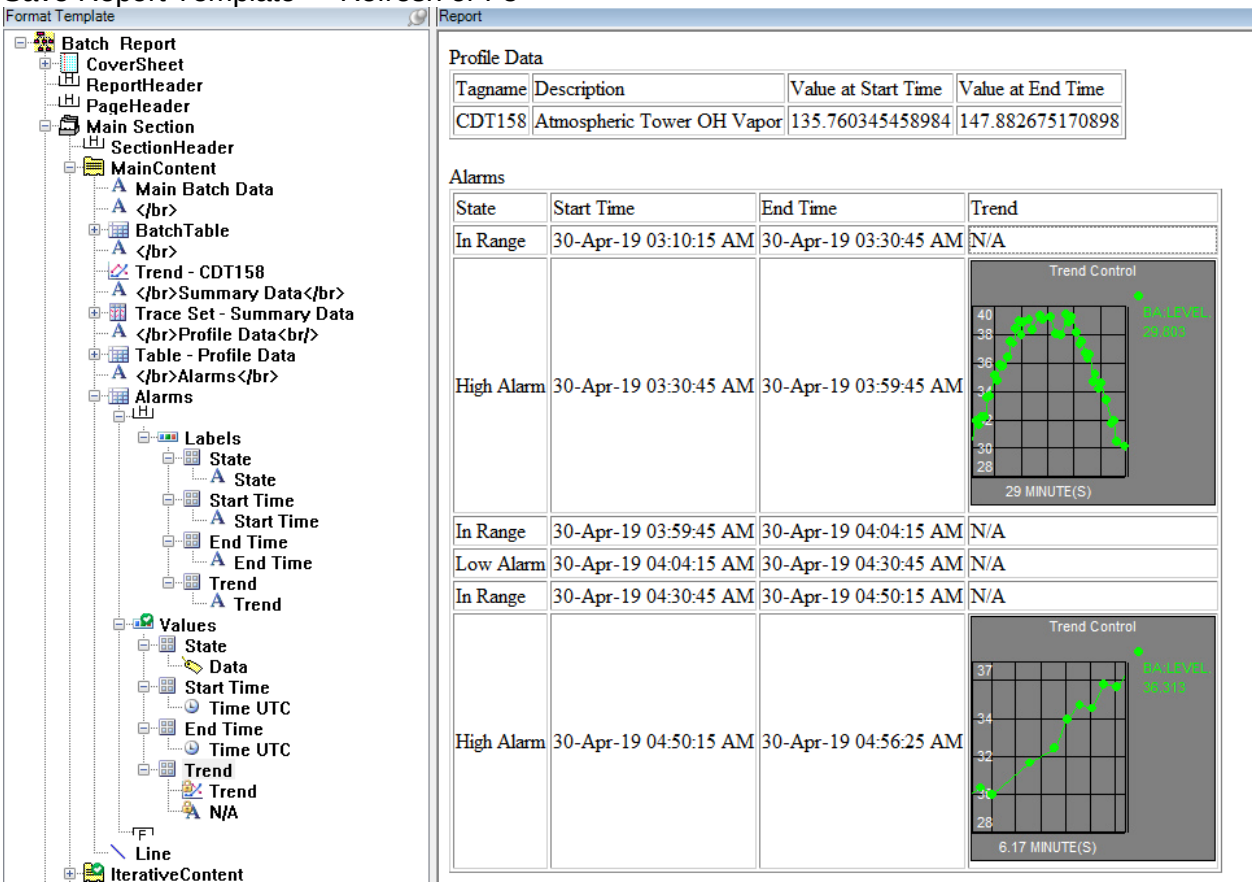
My Alarms

State	Start Time	End Time	Trend
High Alarm	21-Feb-06 01:53:20 PM	21-Feb-06 02:24:20 PM	
Low Alarm	21-Feb-06 02:28:50 PM	21-Feb-06 02:54:50 PM	NA

Solution 9: Using Conditional Formatting

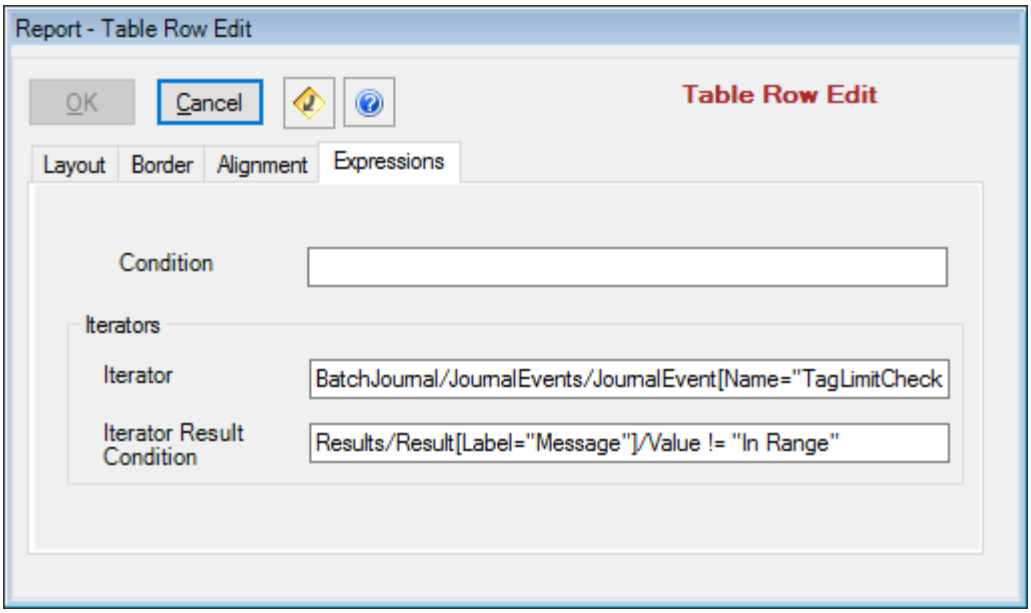
Remove the --- Symbol and replace them with a string of text

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Open the *Batch_Report* template from the previous exercises
3. In the Format Template tab -> expand the “Main Section” -> expand the “MainContent” entry -> expand the “Alarms” table -> expand the “Values” row -> expand the “Trend” column
4. Double clicks the “Trend” object -> Expressions Condition: *boolean(Results/Result[Label="Trend"]/Value)* -> OK
5. Select the “Trend” column of the “Values” row -> Insert -> Text -> Text: N/A -> OK. Notice: lock
6. Double clicks the newly added Text object -> Expressions Condition: *not(boolean(Results/Result[Label="Trend"]/Value)).* Notice: lock
7. Save Report Template -> Refresh or F5

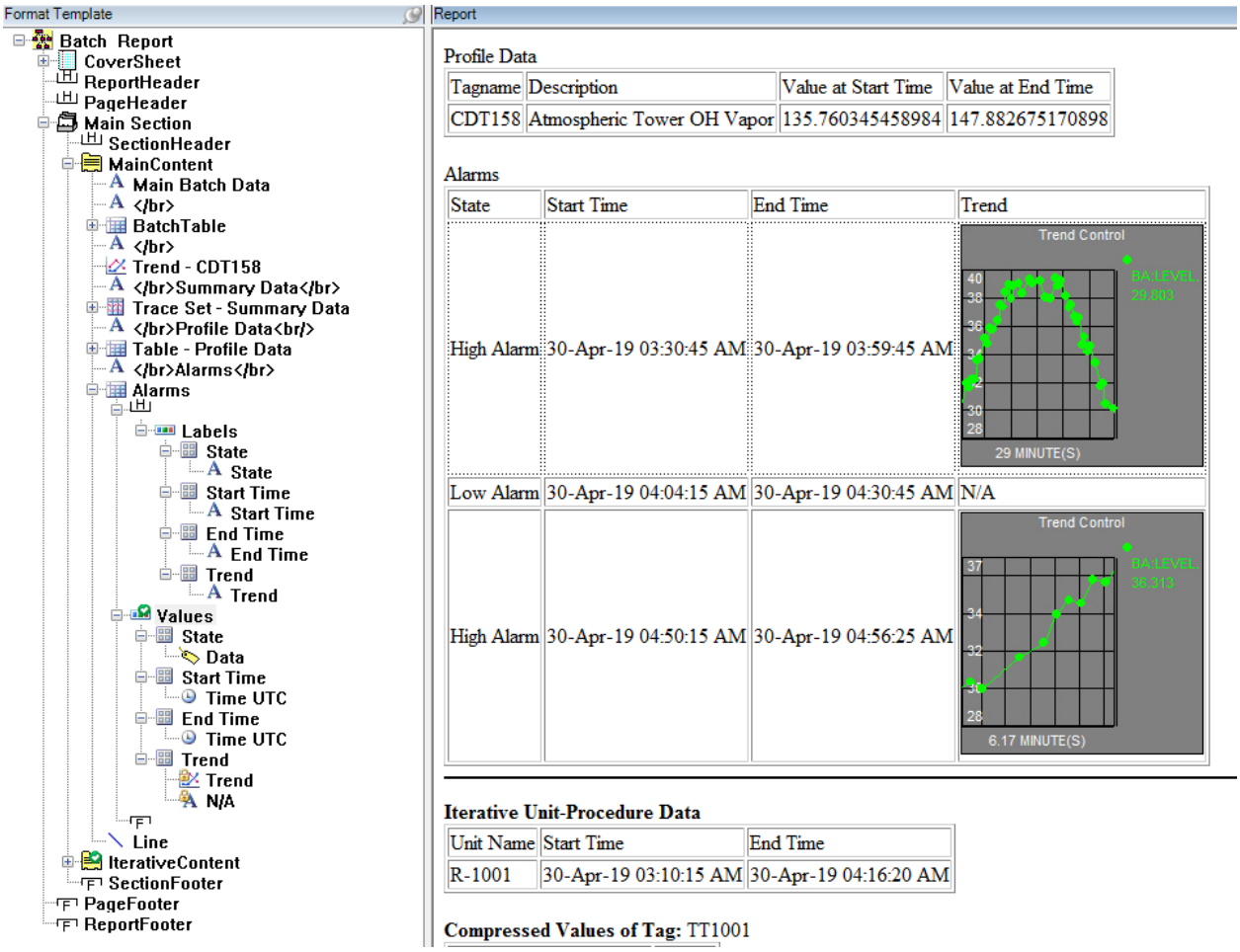


Remove the “In Range” State

1. In the Format Template tab -> expand the “Main Section” -> expand the “MainContent” entry -> expand the “Alarms” table
2. Double clicks the “Values” row of the Alarms table -> Expressions tab -> Iterator Result Condition: *Results/Result[Label="Message"]/Value != "In Range"*



3. Save Report Template -> Refresh or F5


Format Template
Batch Report
CoverSheet
ReportHeader
PageHeader
Main Section
SectionHeader
MainContent
Main Batch Data
</br>
BatchTable
</br>
Trend - CDT158
</br>Summary Data</br>
Trace Set - Summary Data
</br>Profile Data
Table - Profile Data
</br>Alarms</br>
Alarms
Labels
State
State
Start Time
Start Time
End Time
End Time
Trend
Trend
Values
State
Data
Start Time
Time UTC
End Time
Time UTC
Trend
Trend
N/A
Line
IterativeContent
SectionFooter
PageFooter
ReportFooter
Report
Profile Data
Tagname
Description
Value at Start Time
Value at End Time
CDT158
Atmospheric Tower OH Vapor
135.760345458984
147.882675170898
Alarms
State
Start Time
End Time
Trend
High Alarm
30-Apr-19 03:30:45 AM
30-Apr-19 03:59:45 AM
Trend Control
BA:LEVEL
29 MINUTE(S)
Low Alarm
30-Apr-19 04:04:15 AM
30-Apr-19 04:30:45 AM
N/A
High Alarm
30-Apr-19 04:50:15 AM
30-Apr-19 04:56:25 AM
Trend Control
BA:LEVEL
6.17 MINUTE(S)
Iterative Unit-Procedure Data
Unit Name
Start Time
End Time
R-1001
30-Apr-19 03:10:15 AM
30-Apr-19 04:16:20 AM
Compressed Values of Tag: TT1001

Lesson 8: Finishing Touches

Toolboxes

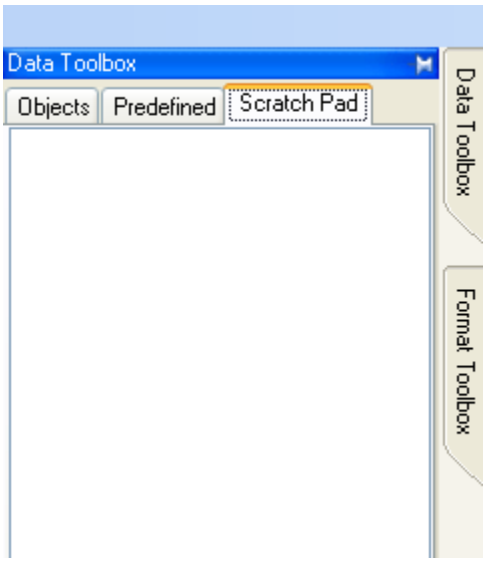
There are two Toolboxes included in the RtReports Editor. These storage areas are used for reusable pieces of formatting or data.

They are accessed from the right edge of the interface. To use any of these items simply drag them out onto the template pages.

Items saved permanently are manually versioned.

ScratchPad

Each has a permanent storage library and a Scratch Pad. The Scratch Pad is temporary. The Scratch Pad will hold items in memory only until the application is closed, and then it is cleared. It is similar to the Clipboard in Microsoft Office.



Data Toolbox

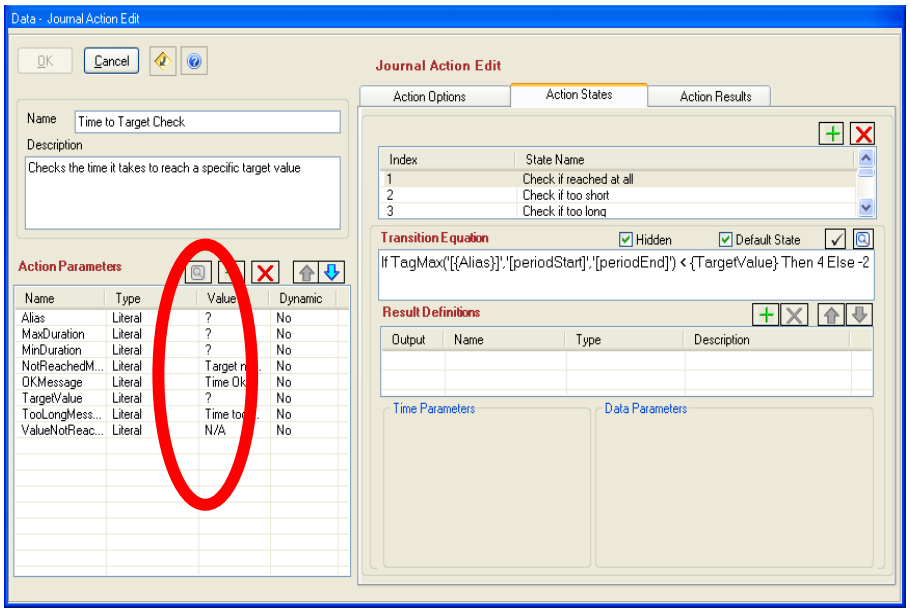
The Data Toolbox holds items that are used in the data template that you want to reuse. The permanent storage area is labeled “Pre-defined.” There are some pre-defined items that ship with the application.

Format Toolbox

The format Toolbox holds items that are used in the report format template that you want to reuse. The permanent storage area is labeled “Style Sheets.”

How to use the Toolboxes

Simply create generic items. For example, users can create a data structure with define journal actions with none of the details filled in.



Report designers only need to drag the item into the Data Template and populate the Action Parameters fields of the Journal Action.

Common ways the Style Sheets are used are report covers, headers, and footers that are designed to meet corporate report standards.

Exercise 10: Using Toolboxes: Predefined Actions and Style Sheets

Objectives

Discover how easy to configure a new report using the *PreDefined Action* and *StyleSheets*.

Learn how to import/export Toolboxes Predefined Action and Style Sheets

Problem Description

You want to add a limit check to your report for a critical parameter. Rather than building a state machine, you want to use the built-in function provided with *RtReports* or that you built yourself.

Approach

Create a new Batch based report template called *PredifinedReport*.

Add *Limit Check* action to the UnitBatch level

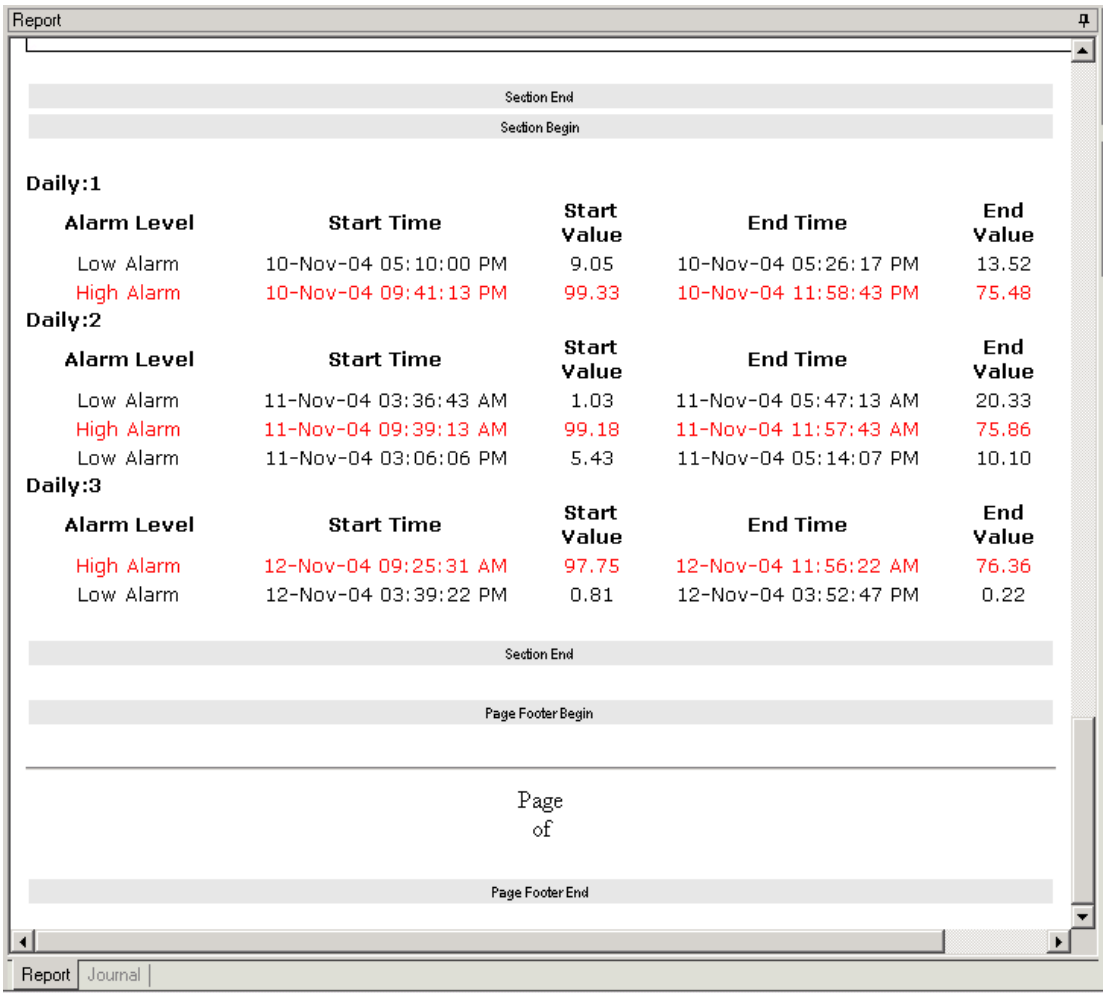
- Add the limit check action “PI Alias Limit Check” from the *Data Toolbox*
- Configure the limit check to point to the critical parameter, the **Level** alias for the unit.
- Configure the high limit to be **68** and the low limit to be **55**.

Add *Limit Check* StyleSeet to your *Format Template*

- Import the limit check action “**Alias Limit Check Section**” into the *Format Toolbox.* Add it to your Format Template.
- Import the cover page, report header, page header and page footer StyleSheets from the appropriate class folder. Add them to your Format Template.

Test the report against a completed EF of the 1K Reactors.

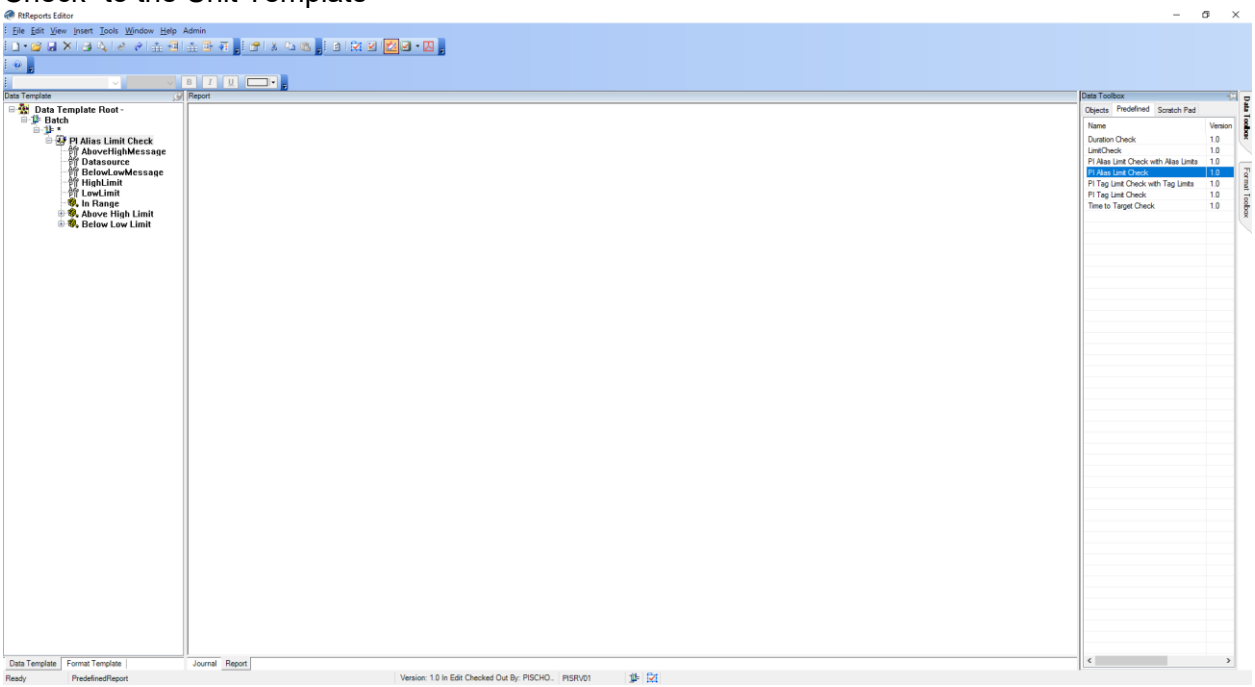
Report sample below:


Report
Section End
Section Begin
Daily:1
Alarm Level
Start Time
Start Value
End Time
End Value
Low Alarm
10-Nov-04 05:10:00 PM
9.05
10-Nov-04 05:26:17 PM
13.52
High Alarm
10-Nov-04 09:41:13 PM
99.33
10-Nov-04 11:58:43 PM
75.48
Daily:2
Alarm Level
Start Time
Start Value
End Time
End Value
Low Alarm
11-Nov-04 03:36:43 AM
1.03
11-Nov-04 05:47:13 AM
20.33
High Alarm
11-Nov-04 09:39:13 AM
99.18
11-Nov-04 11:57:43 AM
75.86
Low Alarm
11-Nov-04 03:06:06 PM
5.43
11-Nov-04 05:14:07 PM
10.10
Daily:3
Alarm Level
Start Time
Start Value
End Time
End Value
High Alarm
12-Nov-04 09:25:31 AM
97.75
12-Nov-04 11:56:22 AM
76.36
Low Alarm
12-Nov-04 03:39:22 PM
0.81
12-Nov-04 03:52:47 PM
0.22
Section End
Page Footer Begin
Page
of
Page Footer End
Report
Journal

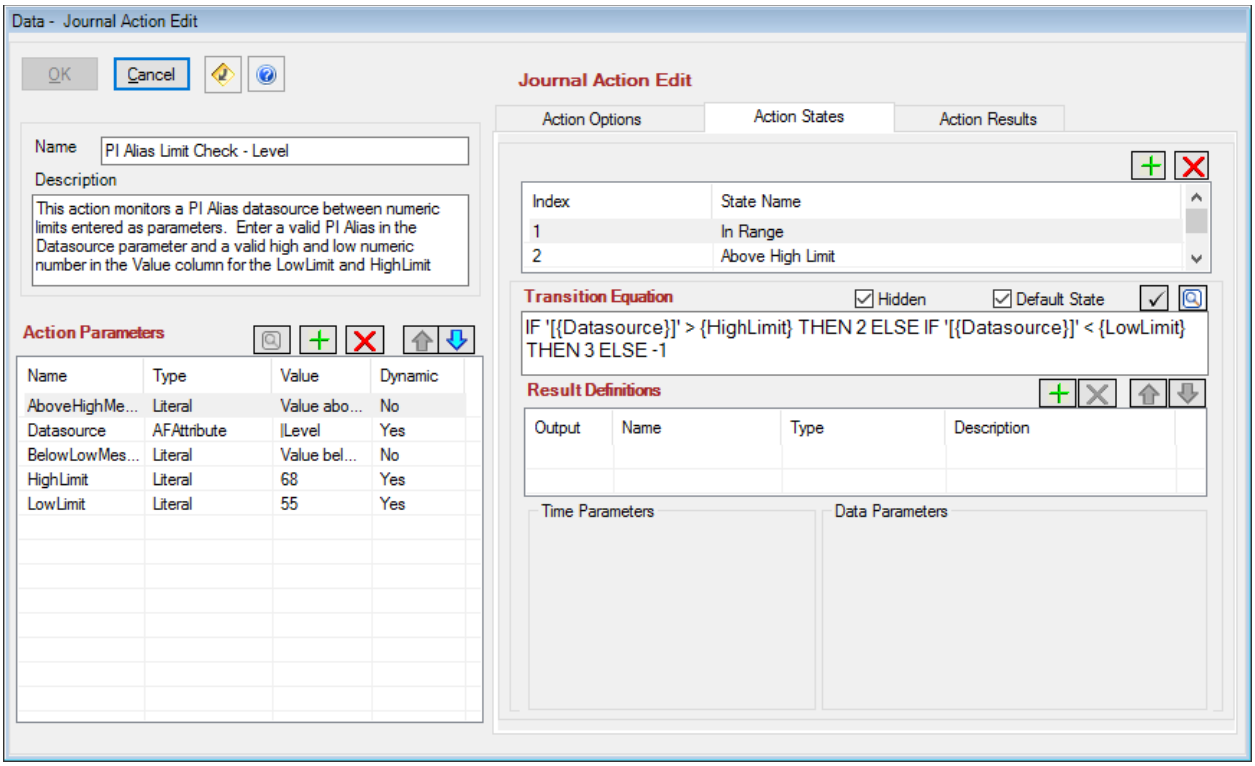
Solution 10: Predefined Actions and Style Sheets

Adding a *PreDefined Action* to *Data Template*

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. New -> Batch Report Template… -> Name: PredefinedReport -> OK
3. In Data Template tab -> select Batch -> Insert -> Unit Template -> OK
4. Click “Data Toolbox” tab (right) -> Predefined tab -> select and drag the “PI Alias Limit Check” to the Unit Template “*”



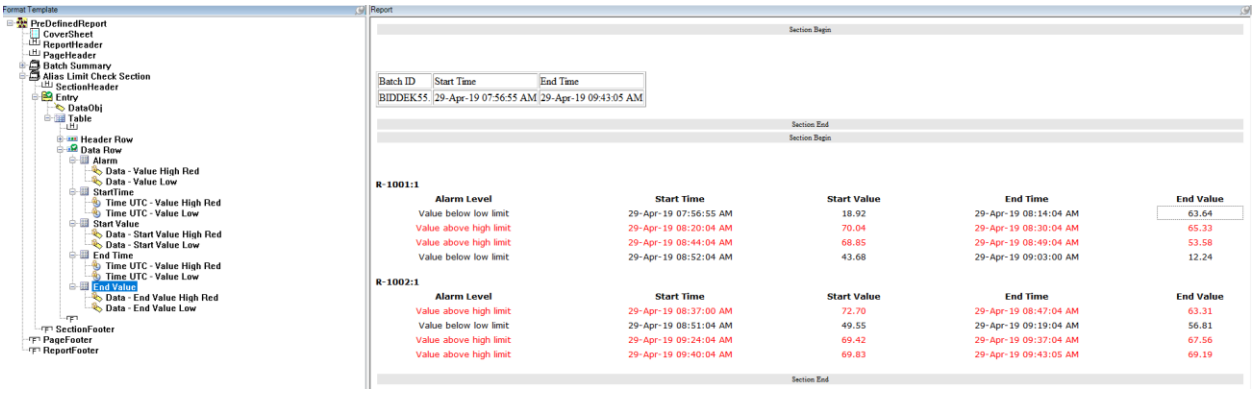
5. Double clicks the “PI Alias Limit Check” Journal Action -> Name: PI Alias Limit Check – Level -> populate Action Parameters:
    - Data Source Type: AFAttribute -> Value: Level (Relative Path of an Element Attribute of the 1K Reactors)
    - HighLimit: 68
    - LowLimit: 55
    - OK



6. Save Report Template
7. View -> Test Report -> Use Event Frames (or F3) -> browse and find a completed EF the 1K Reactors

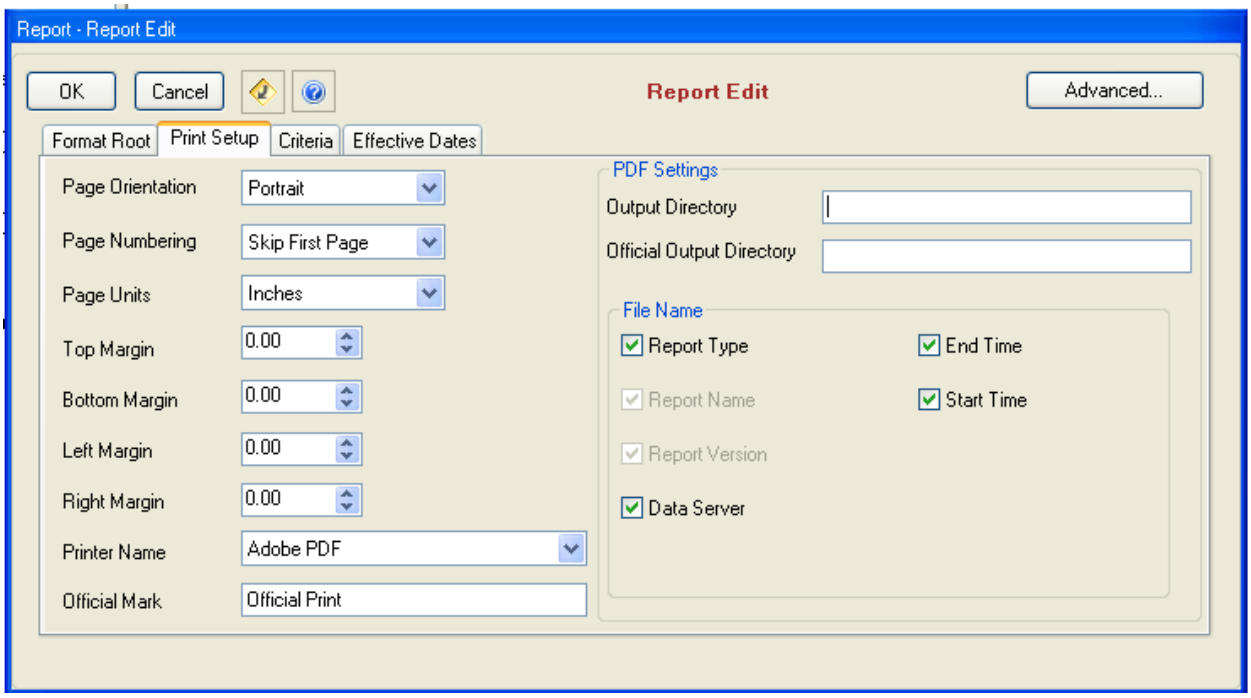
Add Limit Check Style Sheet to Report Contents to display newly configured data.

1. Import a Format Template: File -> Import -> Import Format Template to Editor… -> "C:\Class\Exercises and Solutions\Toolbox Exercise Files\PreDefinedReport_FormatTemplate-Base.XML"
2. Import a Style Sheet: Click *Format Toolbox* tab (right) -> *Style Sheets* tab -> right click anywhere -> *Import* -> "C:\Class\Exercises and Solutions\Toolbox Exercise Files\Alias Limit Check Section.xml" -> Open
3. Import more Style Sheets: repeat importing the Generic Cover Page, Generic Header, Page Header, Page Footer.
4. Switch to *Format Template* tab
5. Click *Format Toolbox* tab (right) -> select and drag *Alias Limit Check Section* to the report root -> re-arrange such that it is below the Batch Summary section
6. Save Report Template -> Refresh (F5)


Format Template
PreDefinedReport
CoverSheet
ReportHeader
PageHeader
Batch Summary
Alias Limit Check Section
SectionHeader
Entry
DataObj
Table
Header Row
Data Row
Alarm
Data - Value High Red
Data - Value Low
StartTime
Time UTC - Value High Red
Time UTC - Value Low
Start Value
Data - Start Value High Red
Data - Start Value Low
End Time
Time UTC - Value High Red
Time UTC - Value Low
End Value
Data - End Value High Red
Data - End Value Low
SectionFooter
PageFooter
ReportFooter
Report
Section Begin
Batch ID
Start Time
End Time
BIDDEK55
29-Apr-19 07:56:55 AM
29-Apr-19 09:43:05 AM
Section End
Section Begin
R-1001:1
Alarm Level
Start Time
Start Value
End Time
End Value
Value below low limit
29-Apr-19 07:56:55 AM
18.92
29-Apr-19 08:14:04 AM
63.64
Value above high limit
29-Apr-19 08:20:04 AM
70.04
29-Apr-19 08:30:04 AM
65.33
Value above high limit
29-Apr-19 08:44:04 AM
68.85
29-Apr-19 08:49:04 AM
53.58
Value below low limit
29-Apr-19 08:52:04 AM
43.68
29-Apr-19 09:03:00 AM
12.24
R-1002:1
Alarm Level
Start Time
Start Value
End Time
End Value
Value above high limit
29-Apr-19 08:37:00 AM
72.70
29-Apr-19 08:47:04 AM
63.31
Value below low limit
29-Apr-19 08:51:04 AM
49.55
29-Apr-19 09:19:04 AM
56.81
Value above high limit
29-Apr-19 09:24:04 AM
69.42
29-Apr-19 09:37:04 AM
67.56
Value above high limit
29-Apr-19 09:40:04 AM
69.83
29-Apr-19 09:43:05 AM
69.19
Section End

Defining Printing

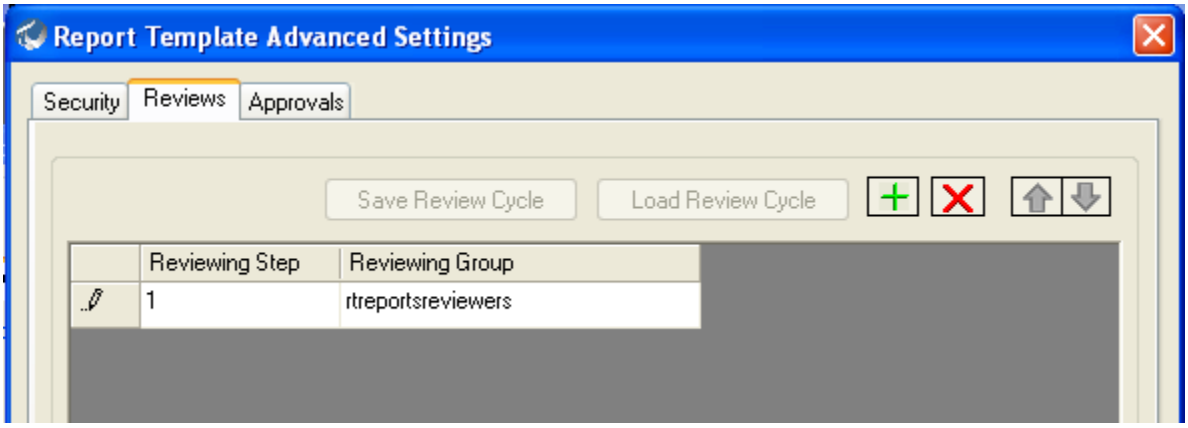
In order to print reports users must to define certain parameters; settings is defined through the Format template Root Properties.



Versioning

All Report Templates have a Version number and Status. The initial version of a Report Template starts at 1.0. It is then incremented by whole numbers. A released Report Template cannot be changed. You must create a new version.

Only one released version of a Report Template may be run against a batch.



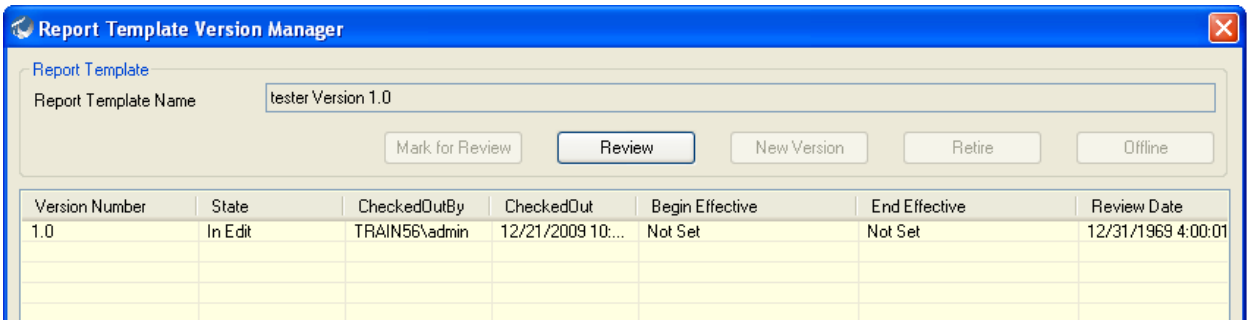
Versioning is handled through the approval cycle. This is accessed through the Format Template Root Properties and selecting “Advanced”.

Note: The Load Review Cycle and Save Review Cycle is not functional.

Reviewing a Report Template with No Review Cycle

To Review a Report Template with No Review Cycle, your group must have the Edit security attribute and must have been granted Edit permission on the specific Report Template. Usually this occurs when the person editing the report has also the ability to release it.

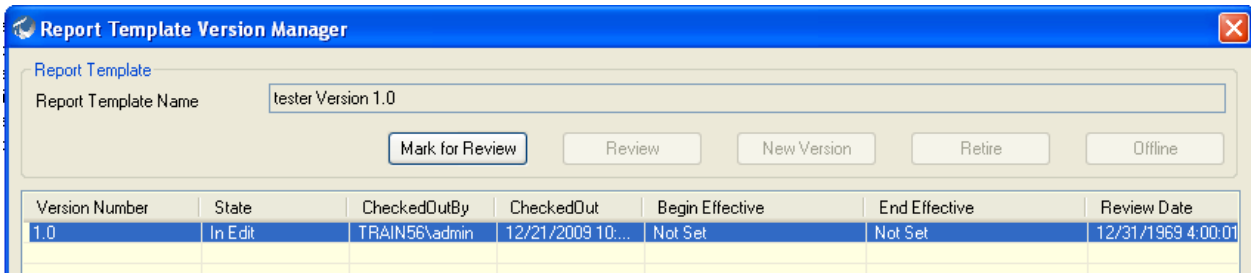
If there are no Approvals configured the report is said to have No Review Cycle. In this scenario the **Mark for Review** button is not used. The initial button will be the **Review** button.



Simply click **Review** to release the report.

Reviewing a Report Template with a Review Cycle

If you are working with a Report Template that has a defined Review Cycle, the first step is to mark the Report Template ‘ready for review.’ To accomplish this, you must have Edit permission on the specific Report Template.

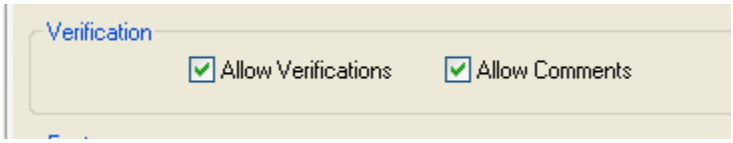


Before the report can be released, it must be set to **Mark for Review**. This means at least one person from each of the groups defined in the Reviews area has to log on and click the **Review** button. When all the Review Groups have clicked **Review** the report is released.

Each reviewer has the capability to enter a comment or reject the report. If the report is rejected the editor must submit it again after making changes.

Allowing Comments and Verifications

In the Format Template Root Properties are check boxes for enabling Verifications and Comments. These are applied to the various sections.



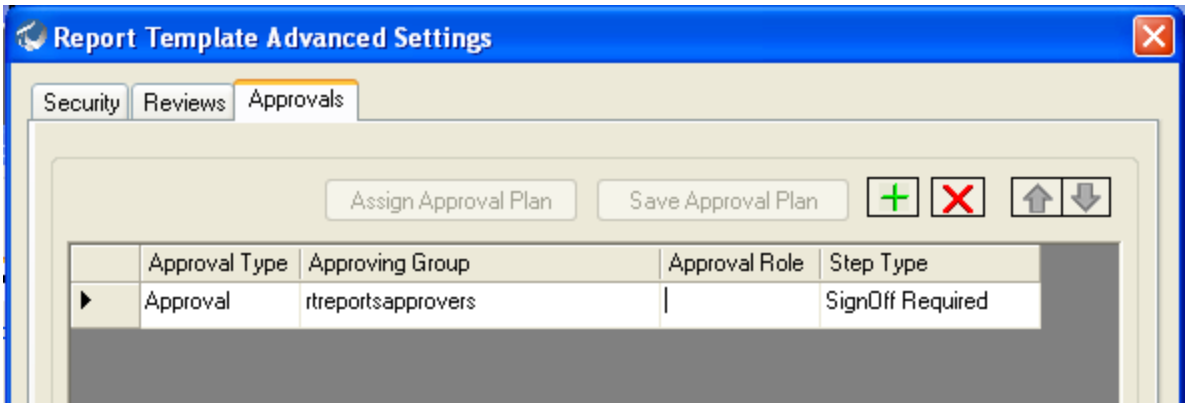
The ability for an individual or group to make comments and verifications is controlled through Group Privileges.

Report Template Version and Status in the Generator

Report Template versions that appear in the Generator depend on three things:

- PIBatch start time
- Report Template Status
- Report Template release date

The Report Template version that is returned is the version which has the most recent release date before the PIBatch start time and has a status of released. If you have several Batches, you may see several versions of a Report Template.



If you have a Report Template checked out for Editing, then this Report Template version is returned as a valid Report Template in the report list that you can see. It does not appear for other users. This is for testing purposes only. When you generate a Report Template that is In Editing, all comments, verification, and approval features are disabled.

Report Status Definitions

Status	Definition
Not Run	The report has not been generated. The batch has started and may not have finished.
Not Yet Approved	The batch has ended. The report has not yet begun its review cycle.
In Approval Cycle	The batch has ended. The report has reached at least one approval person.
Approved	The batch has ended. The report has been approved by all the designated approvers.
In Edit	The report has not been released to Production yet.

Lesson 9: The PI RtReports Generator

In this section we will learn how to generate some basic reports that are predefined. In order to perform these tasks, we will need to import a couple of defined reports.

Import Export Reports

The PI RtReports allows you to do is to export and import report definitions to share between servers or for a backup. Reports are exported and imported in XML format. Each report has two components: Data Template and Format Template. To import a complete report, you must import both pieces.

Exercise 11: Import a Batch-Based Report

Objectives

Quickly create a batch-based report that will later be used with the RtReports Generator. Using the Import function for the Data Template and Format Template.

Problem Description

Given two XML files, import the Data Template and Format Template into a new batch-based report. Once both files are imported, release the newly created report. It will then be available in the RtReports Generator.

Approach

Create a new Batch Report Template called *ReactorReport.*

Import the Data Template file C:\Class\Exercises and Solutions\BatchRecords**ReactorReport_DataTemplate.xml**.

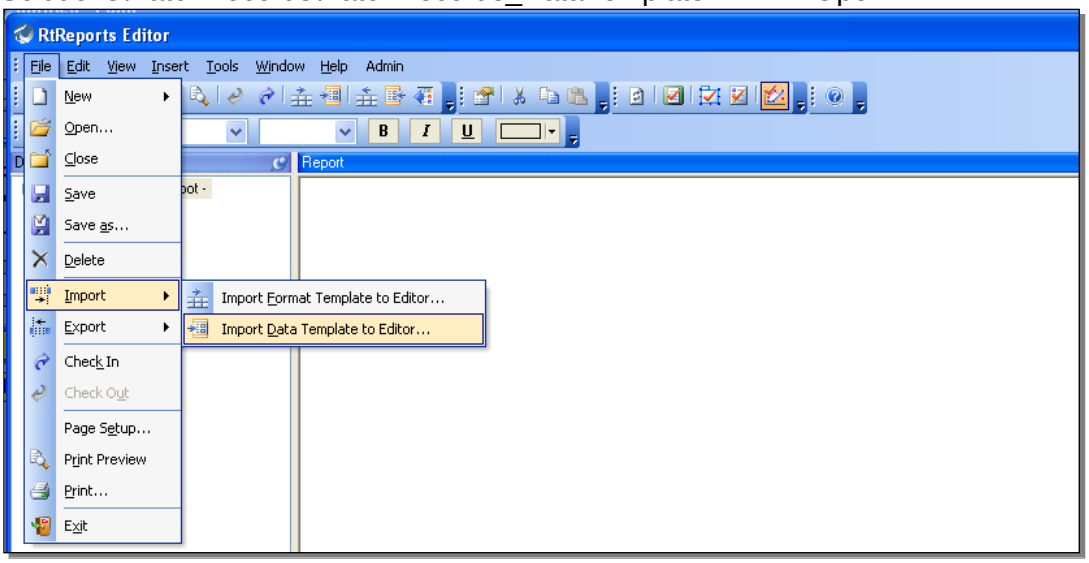
Import the Format Template file C:\Class\Exercises and Solutions\BatchRecords**ReactorReport_FormatTemplate.xml**.

Use the Report Version Manager under the Tools menu to release the report.

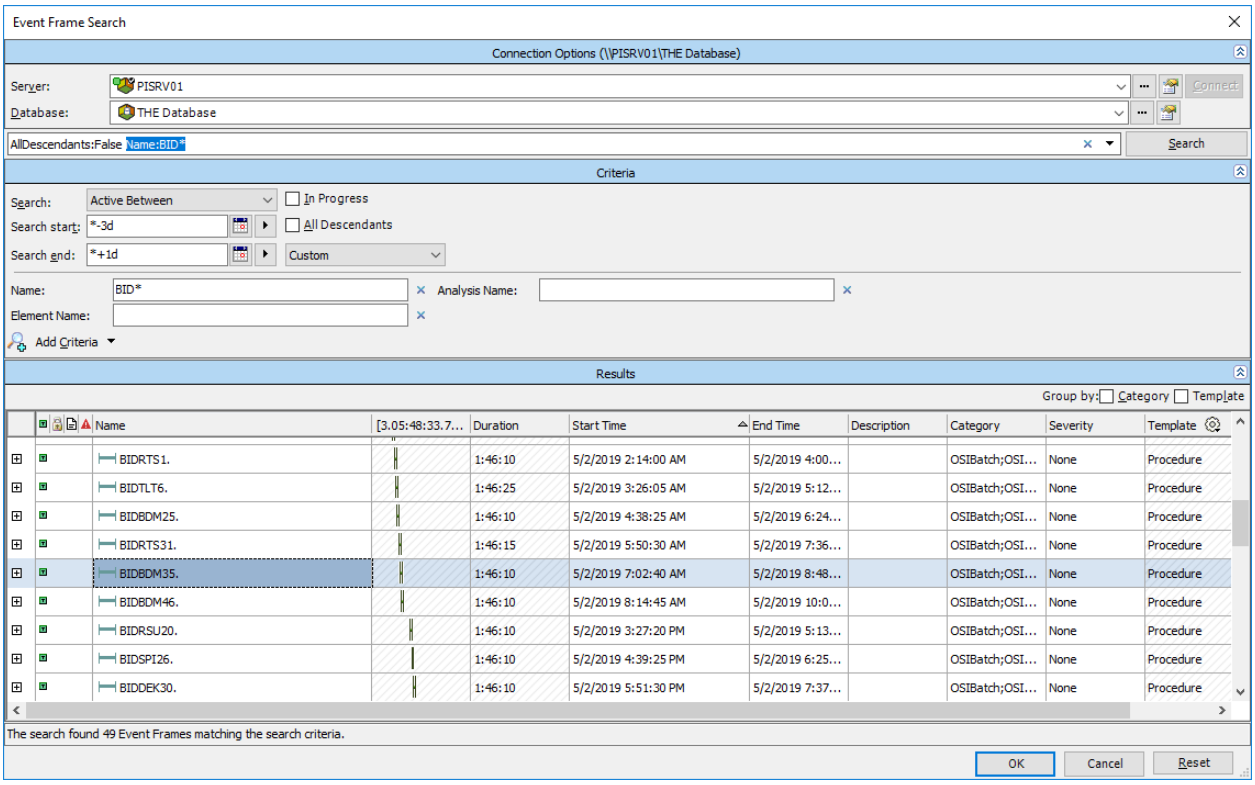
Solution 11: Import a Batch-Based Report

Import a Batch Report Template

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and Login
2. Create a new Batch Report Template: *File* > *New* > *Batch Report Template* -> Name: BatchRecords -> OK
3. Import the Data Template XML file: *File* > *Import* > *Import Data Template to Editor…* -> browse "C:\Class\Exercises and Solutions\BatchRecords\BatchRecords_DataTemplate.xml" -> Open



4. Import the Data Template XML file: *File* > *Import* > *Import Format Template to Editor…* -> browse "C:\Class\Exercises and Solutions\BatchRecords\BatchRecords_FormatTemplate.xml" -> Open
5. Save Report Template -> Vies -> Test Report -
6. Test your report using the *View* > *Test Report* -> *Use Event Frames* (or F3)… -> find and select an 1K Reactors complete "*Procedure*"



7. Select Report tab and verify

Report

OSIsoft

Report Name: BatchRecord
Report Version: 1.0
Start Time: 02-May-19 02:02:40 PM
End Time: 02-May-19 03:48:50 PM
Duration: 1 hour 46 minutes 10 seconds

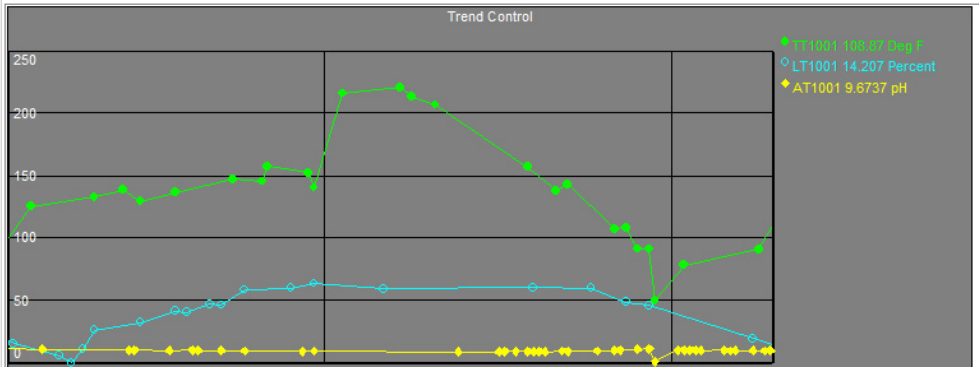
Report Header End

Section Begin

Batch ID:	Start Time:	End Time:	Product	Recipe
BIDBDM35.	02-May-19 02:02:40 PM	02-May-19 03:48:50 PM	PrRTS35.	RCSPI35.

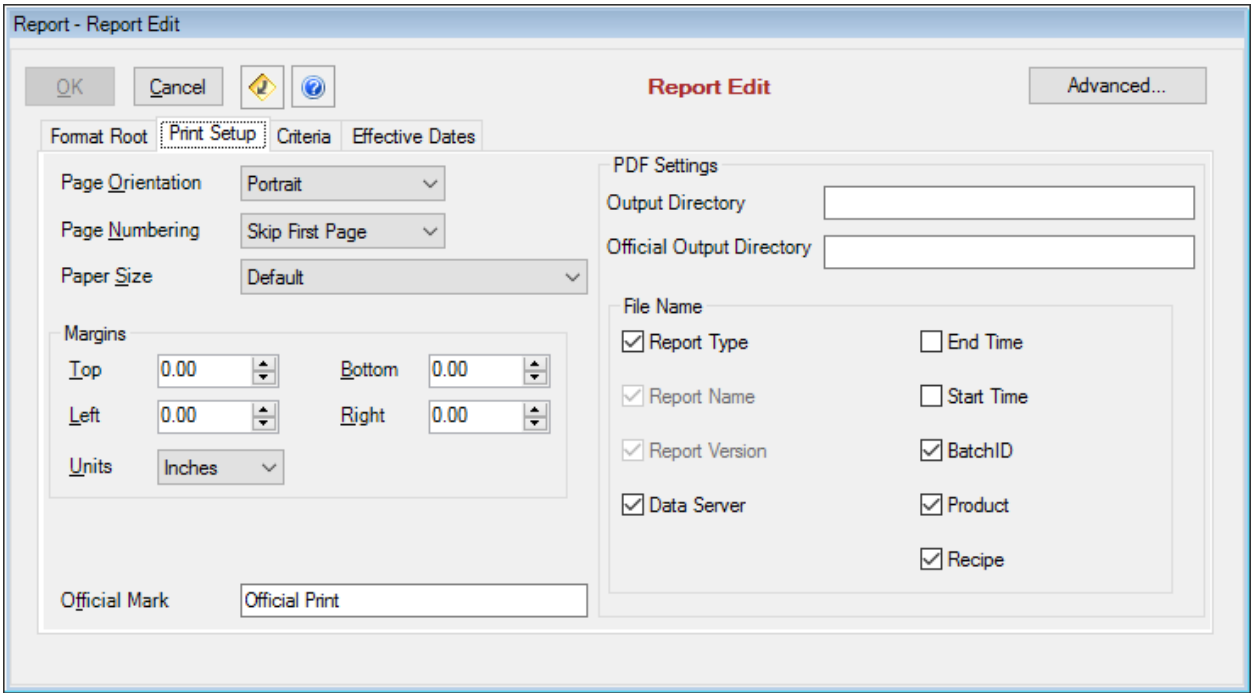
Unit:	Unit Start Time:	Unit End Time:
R-1001	02-May-19 02:02:40 PM	02-May-19 03:08:45 PM

Parameter	Average	Maximum	Minimum	Start Value	End Value
Temperature	142.60	220.48	50.00	100.11	108.87
pH	9.50	10.94	1.00	10.90	9.67
Level	44.84	63.43	0.00	17.34	14.21

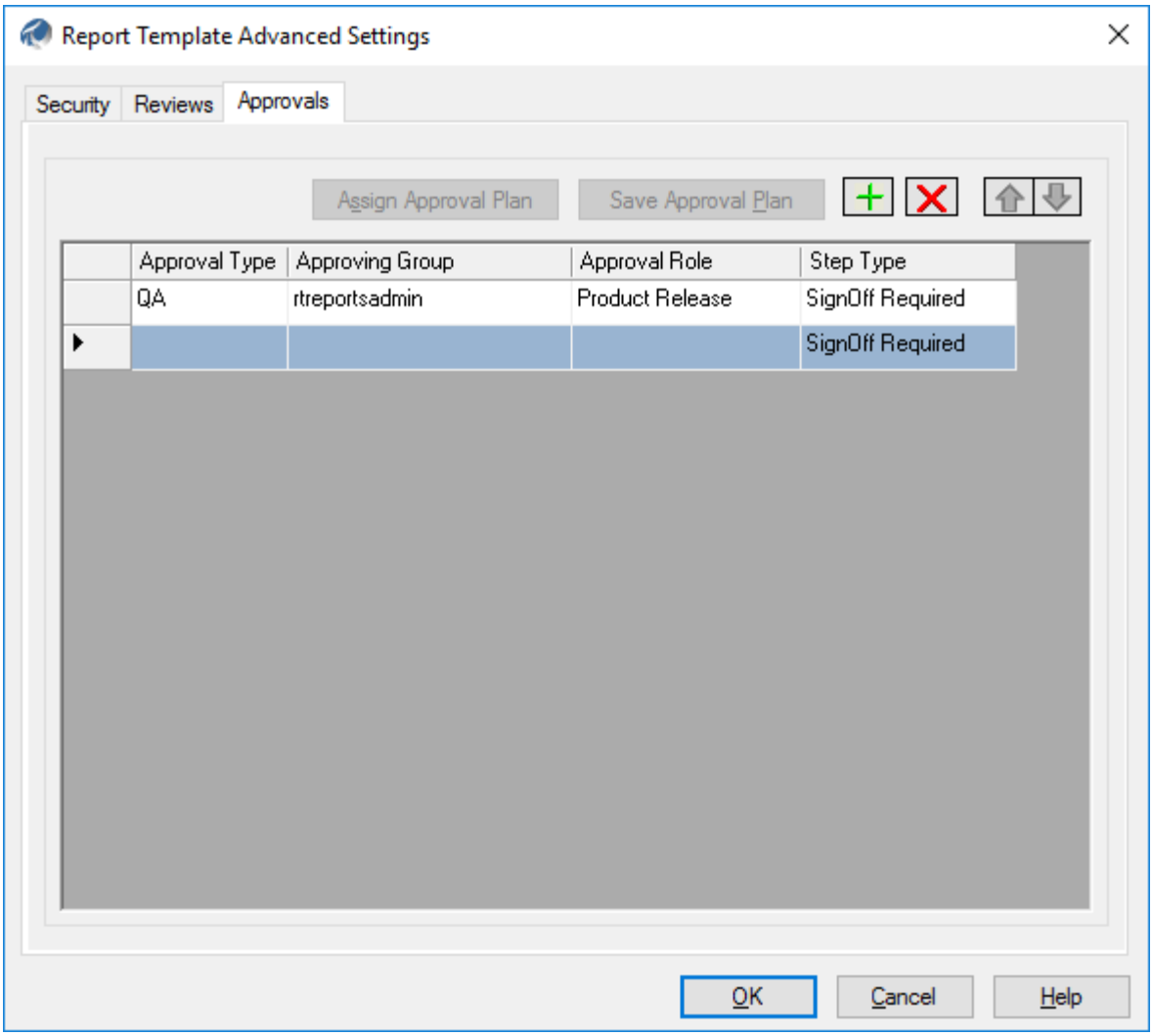


Release a Batch Report

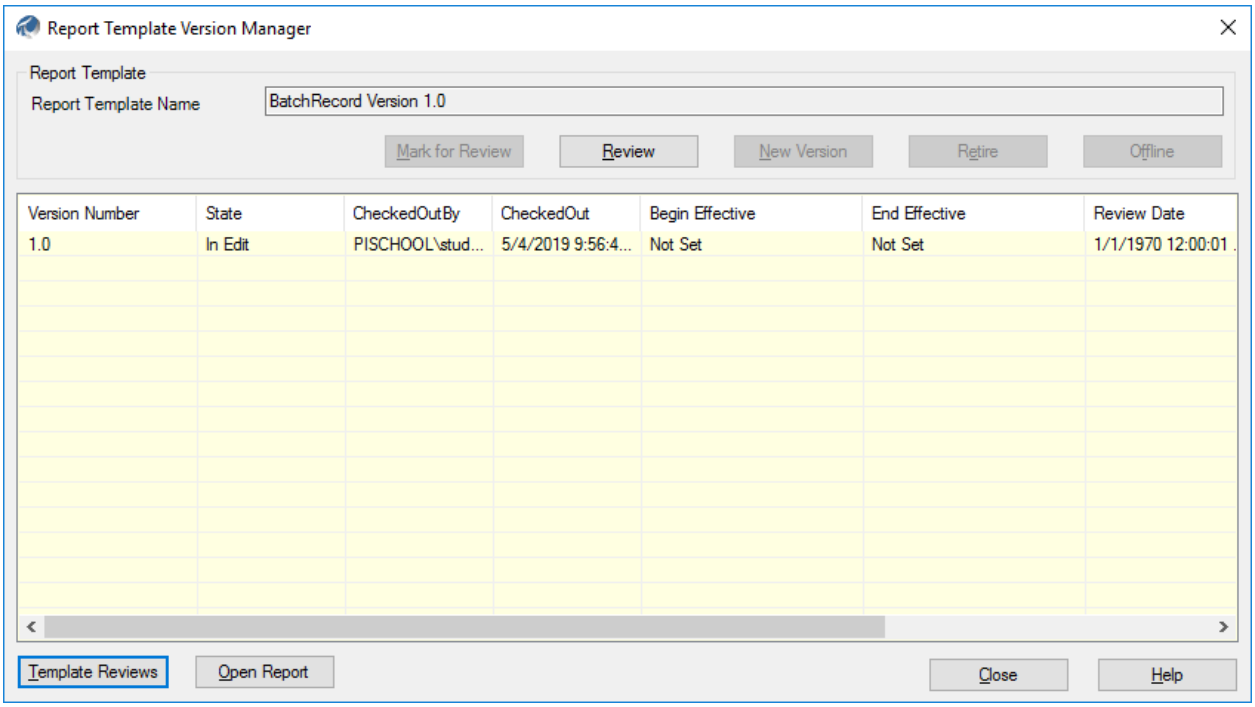
1. Switch to Format Template tab -> verify the report settings as followings:
   - Double clicks report root
   - Format Root tab: check or uncheck *Allow Verification, Allow Comments*…
   - Print Setup tab: *Page Orientation, Page Numbering, Paper Size, Margins*…
   - Criteria tag: *Recipe, Product*…
   - Effective Dates tab: *Effective Start Time* and *Effective Ent Time*



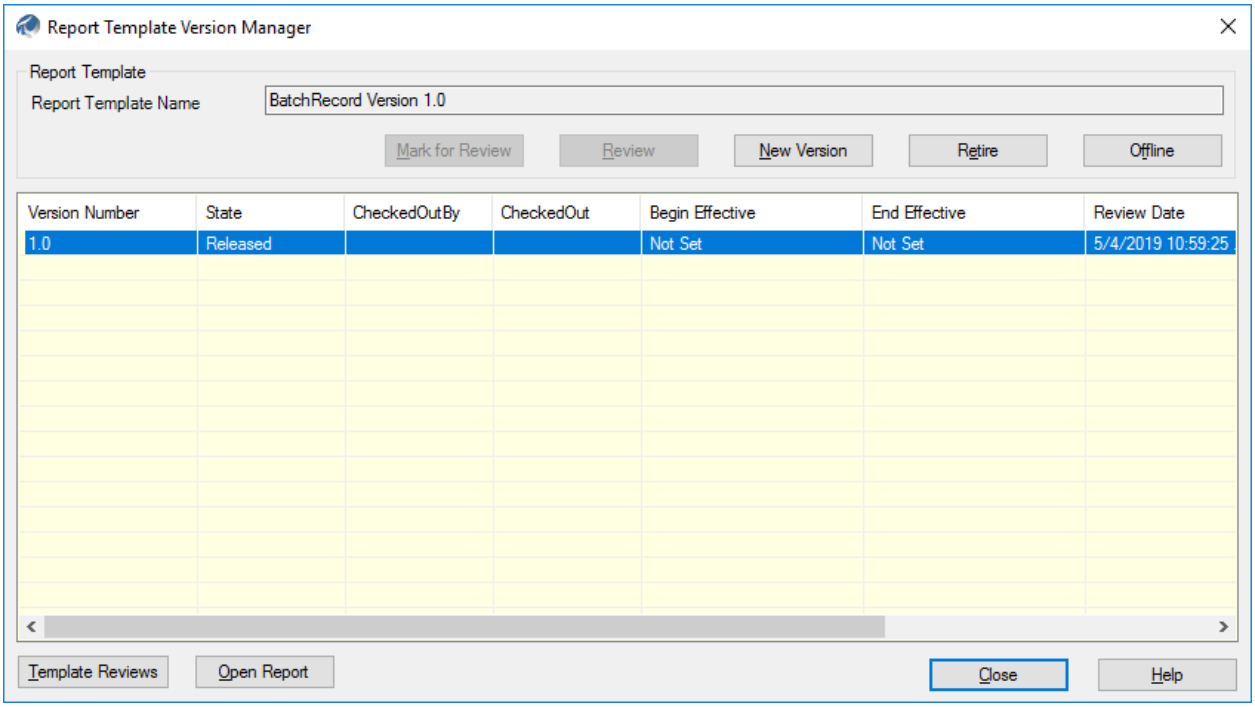
   - Advanced… button
     - Security tab -> setup *Report Edit* Group(s) and *Report Execute* Group(s)
     - Reviews tab -> setup Report Reviewing Step(s) and Group(s)
     - Approvals tab -> setup Report Approval Type(s), Group(s), and Role(s)



2. To release the report: *Tools* > *Report Version Manager…* -> click Review button (to start Review) -> select **Yes** to warning *"This report has no approval cycle. Do you still want to proceed?"* -> Close



3. Notice the State column



4. Click *Close* to complete releasing the report

Exercise 12: Import a Time-Based Report

Objectives

- Quickly create a time-based report that will later be used with the RtReports Generator.
- Using the Import function for the Data and Report Templates.

Problem Description

Given two XML files, import the Data and Format Templates into a new time-based report. Once both files are imported, release the newly created report. It will then be available in the RtReports Generator.

Approach

Create a new Time Report Template called *DailyReport*.

Import the Data Template file C:\Class\Exercises and Solutions\DailyReport**DailyReport_DataTemplate.xml.**

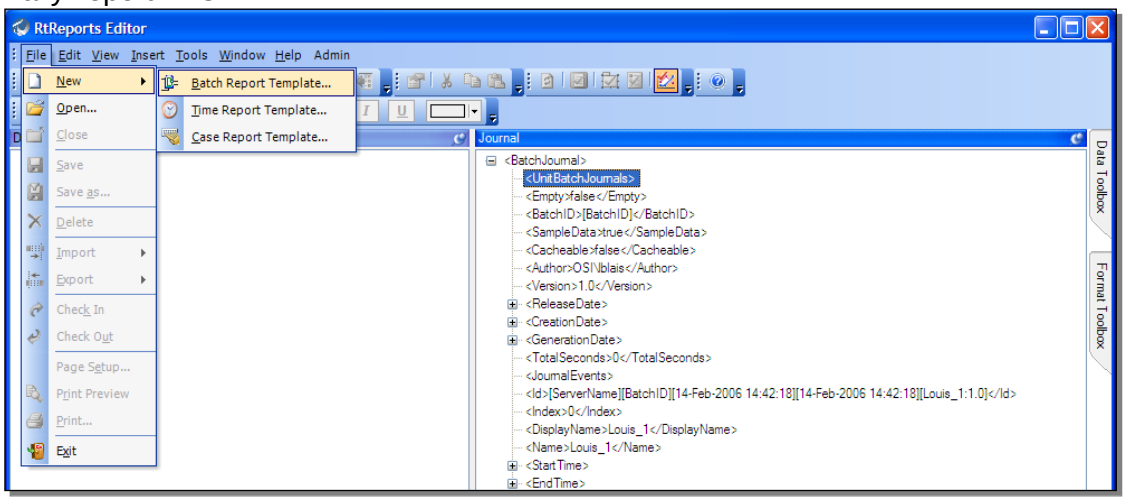
Import the Format Template file C:\Class\Exercises and Solutions\DailyReport**DailyReport_FormatTemplate.xml.**

Use the Report Version Manager under the Tools menu to release the report.

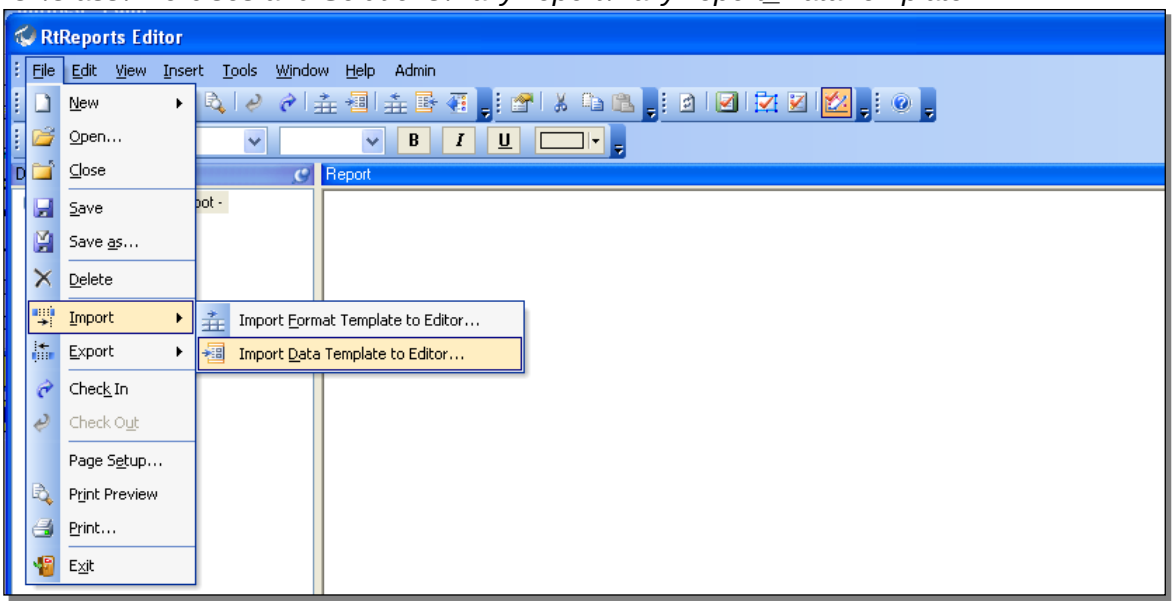
Solution 12: Import a Time-Based Report

Import a Time Report Template

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Create a new Time Report Template: *File > New > Time Report Template…* -> Name: DailyReport -> OK



3. Import the Data Template XML file: *File > Import > Import Data Template to Editor… "C:\Class\Exercises and Solutions\DailyReport\DailyReport_DataTemplate.XML"*



4. Import the Format Template XML file: *File > Import > Import Format Template to Editor… "C:\Class\Exercises and Solutions\DailyReport\DailyReport_FormatTemplate.XML"*
5. Save Report Template -> View -> Test Report… (or F2) -> Data Server: PISRV01 -> Start Time: *-1d -> End Time: * -> OK
6. Report tab -> verify report

Report

Daily Report

Start TIme	End Time
03-May-19 11:35:39 PM	04-May-19 11:35:39 PM

Data Summary

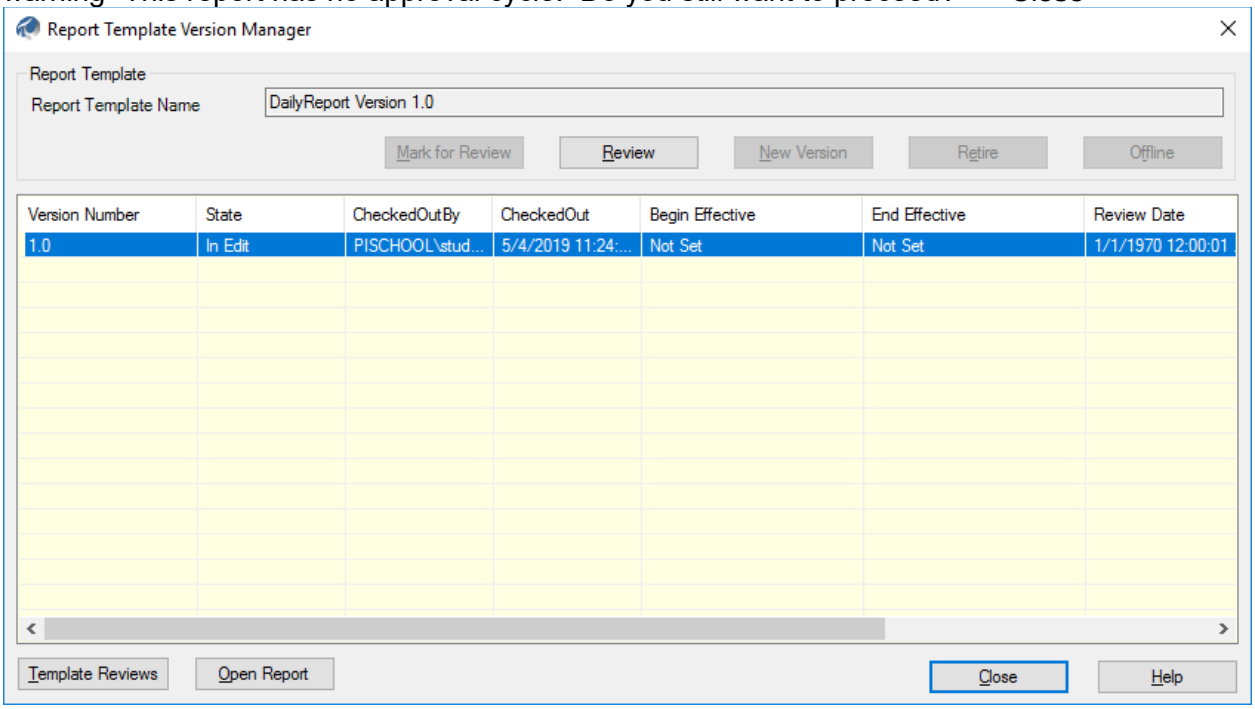
Value	Minimum	Maximum	Average
Atmospheric Tower OH Vapor	47.0826 04-May-19 12:59:01 AM	92.7772 04-May-19 02:23:31 AM	60.8314
Level Reactor 1	0.0000 04-May-19 12:59:01 AM	43.9049 04-May-19 01:48:01 AM	22.2191
Temperature Reactor 1	0.0000 04-May-19 12:59:01 AM	52.4452 04-May-19 11:04:58 PM	18.1636
Concentration Reactor 1	0.0000 04-May-19 12:59:01 AM	46.5248 04-May-19 02:11:01 AM	21.3580

Section End

Section Begin

Release a Time-Based Report

1. Switch to Format Template tab -> verify the Report Setings…
2. To release the report: Tools -> Report Version Manager… -> Review -> Yes to confirm warning “This report has no approval cycle. Do you still want to proceed?” -> Close



The RtReports Generator

The RtReports Generator is a .NET web application.

Users with appropriate permission login to RtReports generator through a web-browser like Google Chrome, Microsoft Edge, Microsoft Internet Explorer… using his/her Windows credential.

The RtReports Generator URL: https://<RtReportHostname>/RtReports. Users may find the URL pre-defined in the PI System menu.

Once a user login to the RtReports Generator successfully, user will need to choose the type of report (Batch-based or Time-based) to run. The RtReports installation default: Batch-based. Users may choose to change the initial login page preference by changing the value of the app key *<add key="InitialPage" value="Batch" />* under the group <appSettings>. The valid value is “Batch” or “Report”.

Users may choose to launch the RtReports in a time-zone that is differ from PI System’s default. The URL for launch the RtReport Generator in EST: https://<RtReportsServerHostname>/RtReports/webpages/logins/logon.aspx?tz=Eastern Standard Time

To enable the Time-zone functionality, user must edit the value of the <appSettings> <add key="ReportTimeZone" value="Server" /> in the %PIHOME%\RtReports\web.config file. Change the value=”Server” to value=”Client”

Generating Batch Reports

The two steps involved:

1. Batches tab: Search and select the desired batch
2. Switch to Reports tab: Find and select (by clicking on) the desired report template

Once the Batch is selected, switch to the “Reports” tab and user will be presented with a list of report templates that can be used against that Batch. User will not see all reports or all versions of each report according to the Batch Report Rule.

The Batch Report Rule: A PI Batch can only be included in one version of a report template.

Note: the over-riding principle in RtReports is that there is one version of the truth. Once user run a report against a Batch that is remembered as “the version” for that Batch and that will be followed. User cannot run a report, modify a template, and run the report again against that same Batch. That would potentially produce two versions of official reports which is not allowed.

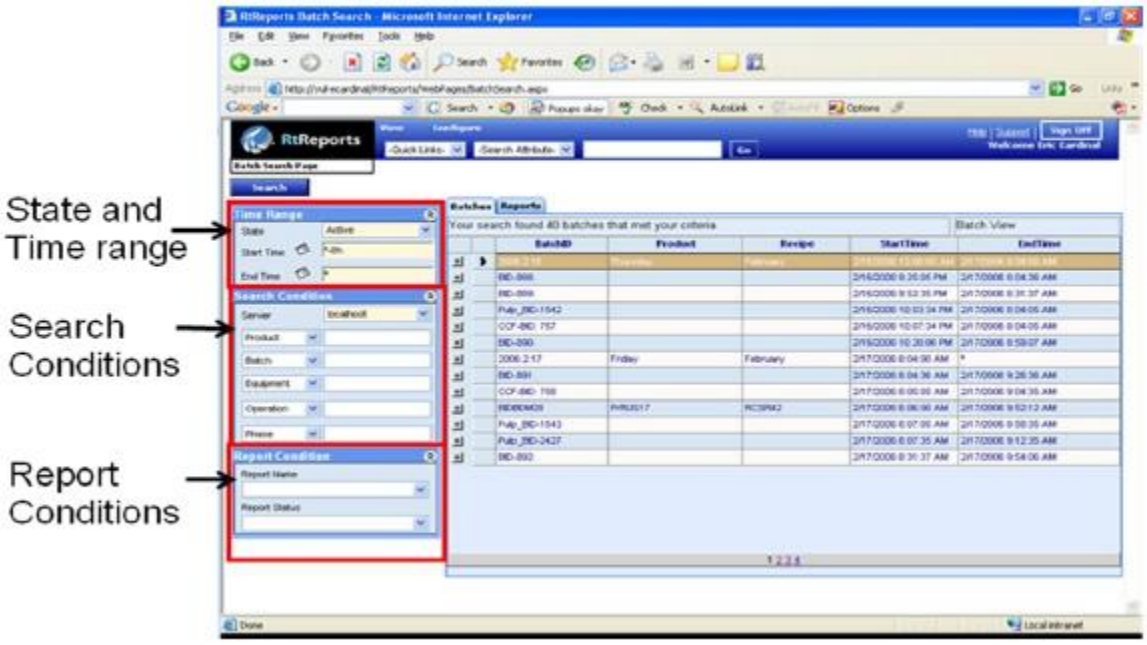
Searching for Batches

Within the **Time Range** group, specify the following:

State:

- Active – Batches that were active between the start and end time specified but not did not occur entirely within the time period.

- Entirely – Batches that started and ended between the start and end time specified.
- Ended – Batches that ended between the start and end time specified.
- Started – Batches that started between the start and end time specified.



Start Time: this time may be relative or absolute.

Start Time: this time may be relative or absolute.

Within the **Search Condition** group, specify the followings:

Data Server: the PI Data Archive server

Asset Server: the PI AF Server

Asset Database: the PI AF Database

Product, Batch, Equipment, Operation, and Phase (user may change the order by clicking on the dropdown arrow): may enter a complete name or part of a name with wildcard characters. The "*" character is treated as a series of wildcard characters. The "?" character is treated as a single wildcard character. Any blank input, the RtReports assumes 'all.'

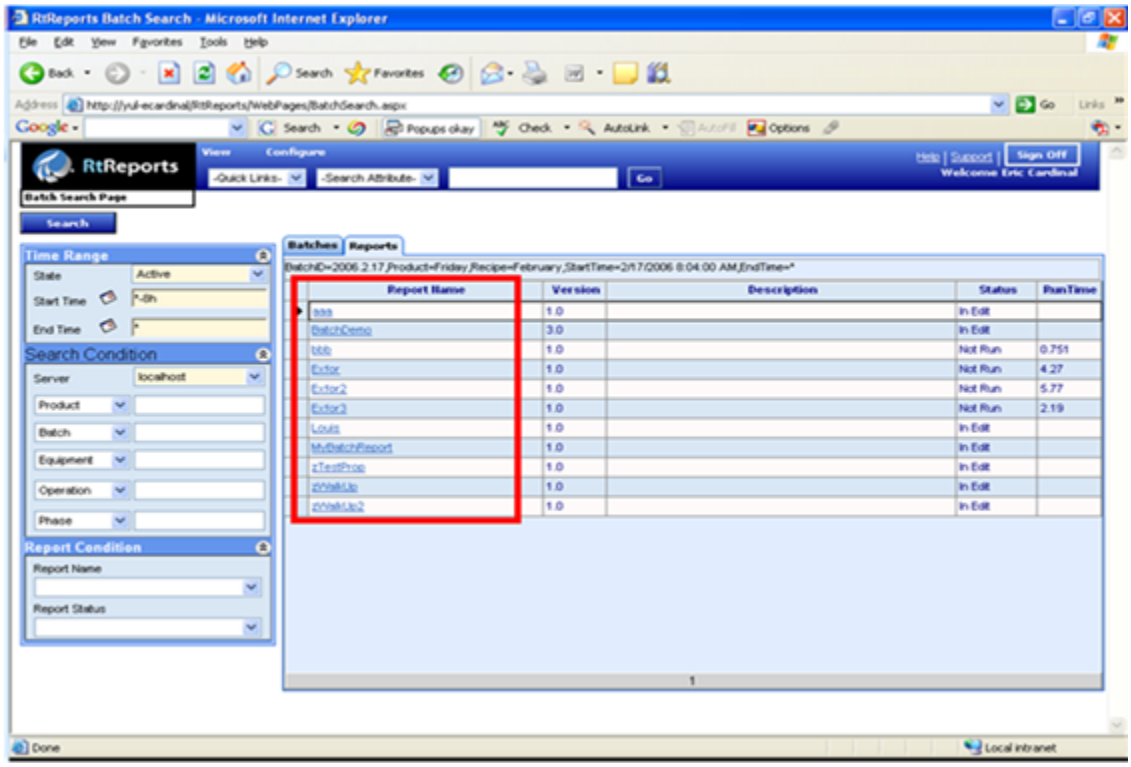
Within the **Report Condition** group, specify the followings:

Report Name: to limit the search to a certain report name

Report Status: to limit the search to a given status for that report

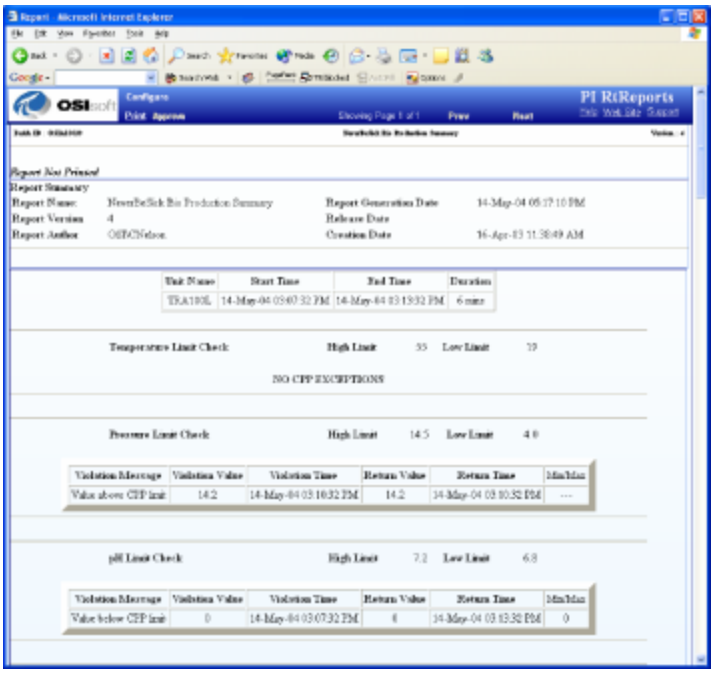
Selecting a Batch Report Template

Once a batch(s) is selected, switching to the “Reports” tab will show the templates that can be used for the Batches selected in the “Batches” tab. To generate the report, simply click on the name of the desired report template.



Viewing the Report

The report will appear in a pop-up window.



There are many options at this point, depending on what has been configured.

NOTE: testing a report template, i.e. generate a report against a non-released report template, the options for Comments, Verifies, Approve, and Print (official print) will not be available to user.

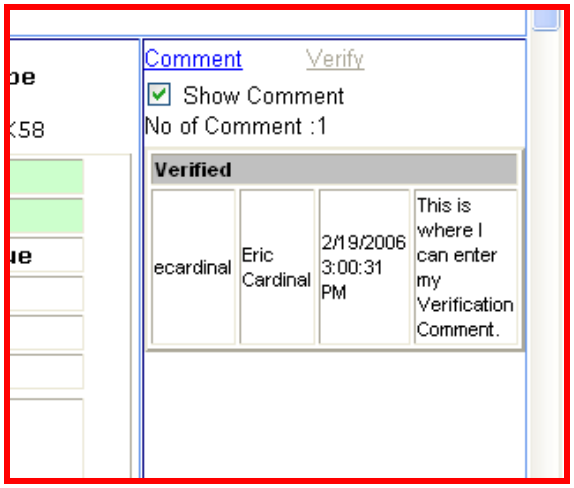
Workflow Options

Viewing a report that is yet to be printed, user with the appropriate assigned permission(s) may perform the optional workflow actions (Comment, Verify, and/or Approve) that are enabled to the report template

Adding Comments & Verifications

If the Comments option is enabled on the report template, the *Comment* link will be enabled to the login user who has the Comments permission. To enter comment(s), click on the *Comment* link, enter comment, and sign-off to complete the action.

If the Verifications option is enabled on the report template, the *Verify* link will be enabled to the login user who has the Verifications permission. To enter verification(s), click on the *Verify* link, enter verification, and sign-off to complete Action.



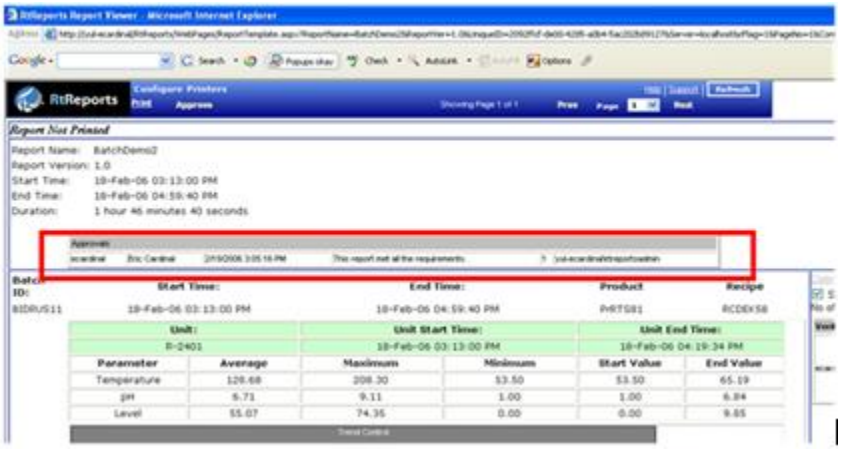
Notice that in the Comment or Verify popup, the User Id is the login Username and cannot be altered. The Username and password combination of authentication satisfies the requirement of Electronic Signatures for 21 CFR Part 11.

Approval

The login user who has the *Approve* permission can only approve a *competed batch* report (i.e. batch that has start time and end time), hence the *Approve* link is enable. The *Approve* link only available to the current approval cycle group and stay available until the group signoff on the Approval.

The RtReports uses its security mechanism to automatically detect what individuals can approve reports. **A Batch must be completed (It must have a Batch End time.) before approvals can occur**. More than one approval may be required.

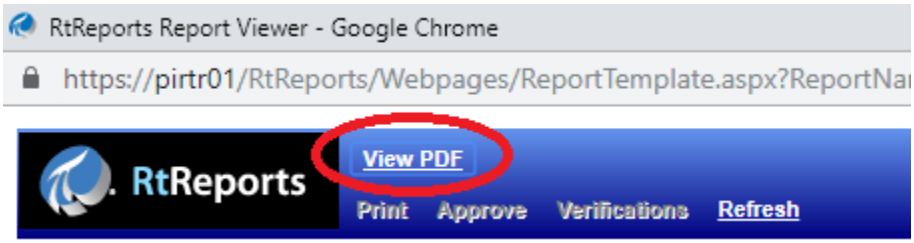
Also notice that in the Approve popup, the User Id is the login Username and cannot be altered. The Username and password combination of authentication satisfies the requirement of Electronic Signatures for 21 CFR Part 11.



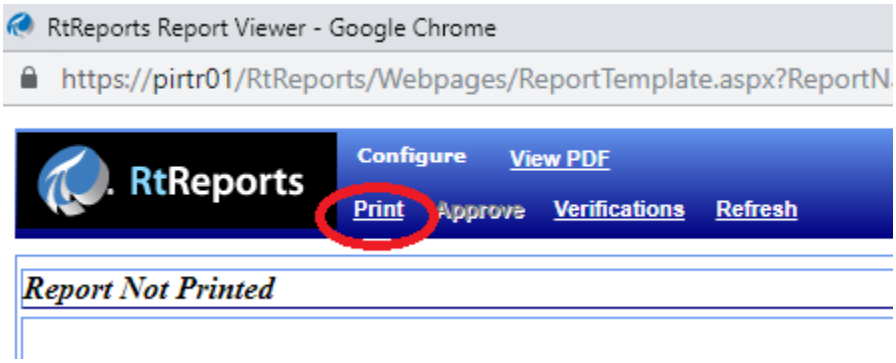
Approval text will be displayed in an approval section of the report under the Report Header Section.

Printing the Report

The two types of RtReports report printing are Official Printing and regular printing. The login user must have the *OfficiallyPrint* permission to initiate Official Printing. Same security rule, the login user must have the *Print* permission to initiate Regular Printing.



Regular Printing is initiated through the View PDF link in the Report Viewer, a .pdf report file will be created in the "AdobePDFLocation" directory (user defined path in %PIHOME%\RtReports\bin\OSIsoft.RtReports.RemotingHost.exe.config file).



Regular Printing is initiated through the View PDF button in the Report Viewer, a .pdf report file will be created to the "OfficialAdobePDFLocation" directory (user defined path in %PIHOME%\RtReports\bin\OSIsoft.RtReports.RemotingHost.exe.config file). Each page of the report has an "Official Print" watermark in the lower left corner of the Report.

Configure automatic report generation

Automatic Report Generation (ARG) is an IIS PreLoad service that runs under the RtReports application pool on the web server.

Edit the %PIHOME%\RtReports\web.config file

- <add key="AdobePDFLocation" value = "<Valid Path>" />. **Note:** the "<Valid Path> must be the same as %PIHOME%\RtReports\bin\OSIsoft.RtReports.RtRemotingHost.exe.config file's
- <add key="ARGSenderEmail" value="<Valid Sender email address>" />
- <network host="<Valid Company SMTP Hostname>" under the <mailSettings> section

Set the RtReports Application Pool User Account

- Open the Internet Information Services (IIS) Manager
- Set Identity to a Domain Service Account. **Note:** The Domain Service Account must be member of the RtReportsAdmin local Group, must have Read/Write access to the %PIHOME%\RtReport, and must have Read/Write access to the RtReports cache folder (default: %PIHOME%\RtReports\cache)
- Set the RtReports App Pool's Idle Time-out (minutes) = 300 (minimum recommended). Right click RtReoirts App Pool -> Advanced Settings… -> **Idle Time-out (minutes)** under Process Model section

Edit the ApplicationHost.config file

- Prior to editing, be sure to save file C:\Windows\System32\inetsrv\config\applicationHost.config
- Set <application path="/RtReports" applicationPool="RtReports App Pool" **serviceAutoStartEnabled="true"** serviceAutoStartProvider="ARGService">

Configure and Enable Automatic Report Generation

- RtReports Generator -> login -> -Quick Links- -> AutomaticReportGeneration -> Go -> Generation Settings section -> Name: … -> check Enabled -> Report Name: … -> Email Recipients: … -> Matching Criteria… -> Save
- To disable any Enabled Automatic Report Generation row ->select the row -> Generation Settings -> uncheck Enabled -> Save
- To delete any Enabled Automatic Report Generation row -> click "X" (right of row)

Exercise 13: Generating a Batch Based Report

Objectives

Given a PI Batch and a Batch-Based report template, use the RtReports Generator to create a new report and use the Comment and Verify options.

Problem Description

Generate a batch report using the *ReactorReport* report template created in an earlier exercise. Add comments and verification comment in one or many sections using the functions on the right-hand side of each section.

Approach

Open the *ReactorReport* template in the RtReports Editor. Validate that the report has comments and verifications enabled.

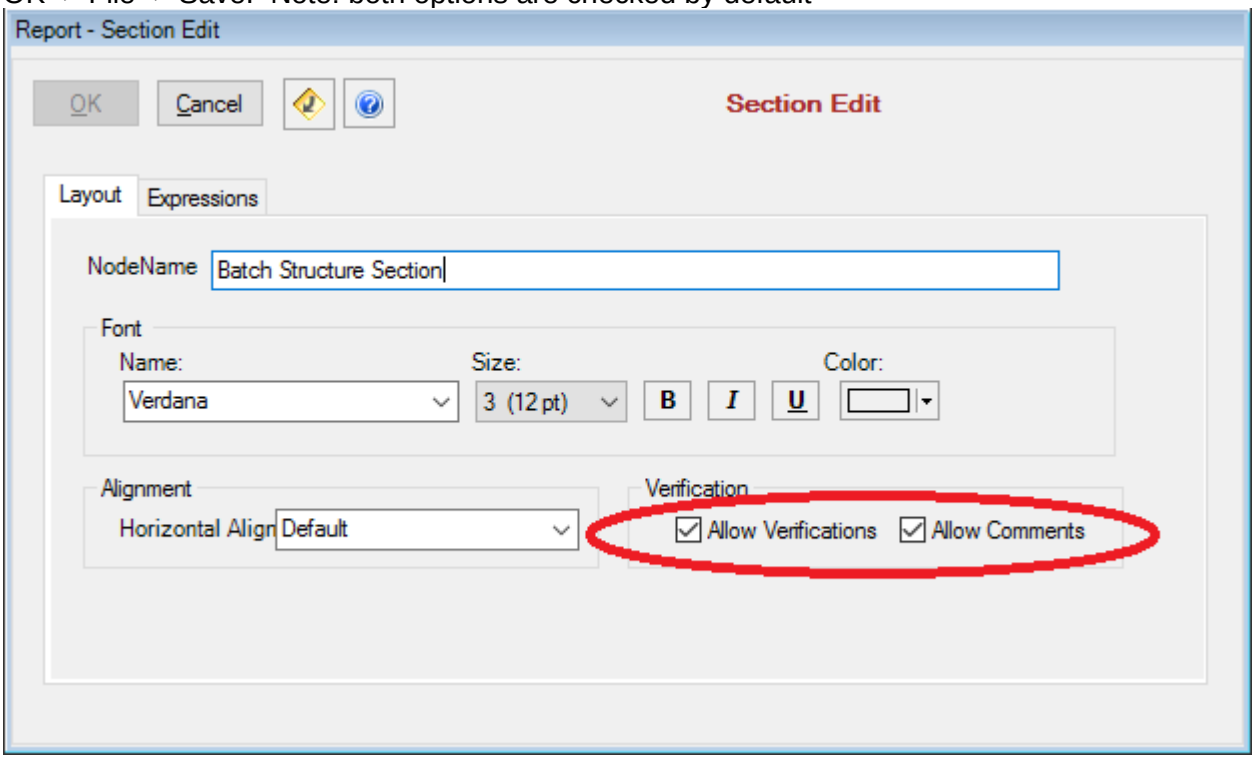
Release the report.

Launch the RtReports Generator and login. Select a completed 1K Reactors Batch and then the *ReactorReport* to generate the report.

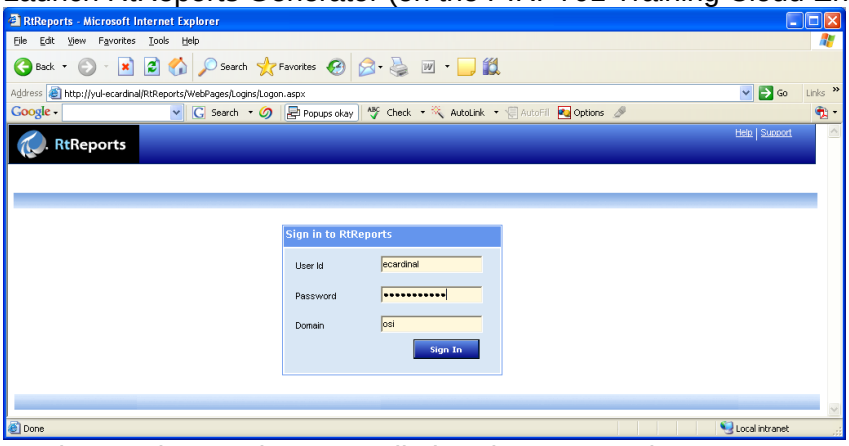
After it is generated, add a comment and verify the report.

Solution 13: Generating a Batch-Based Report

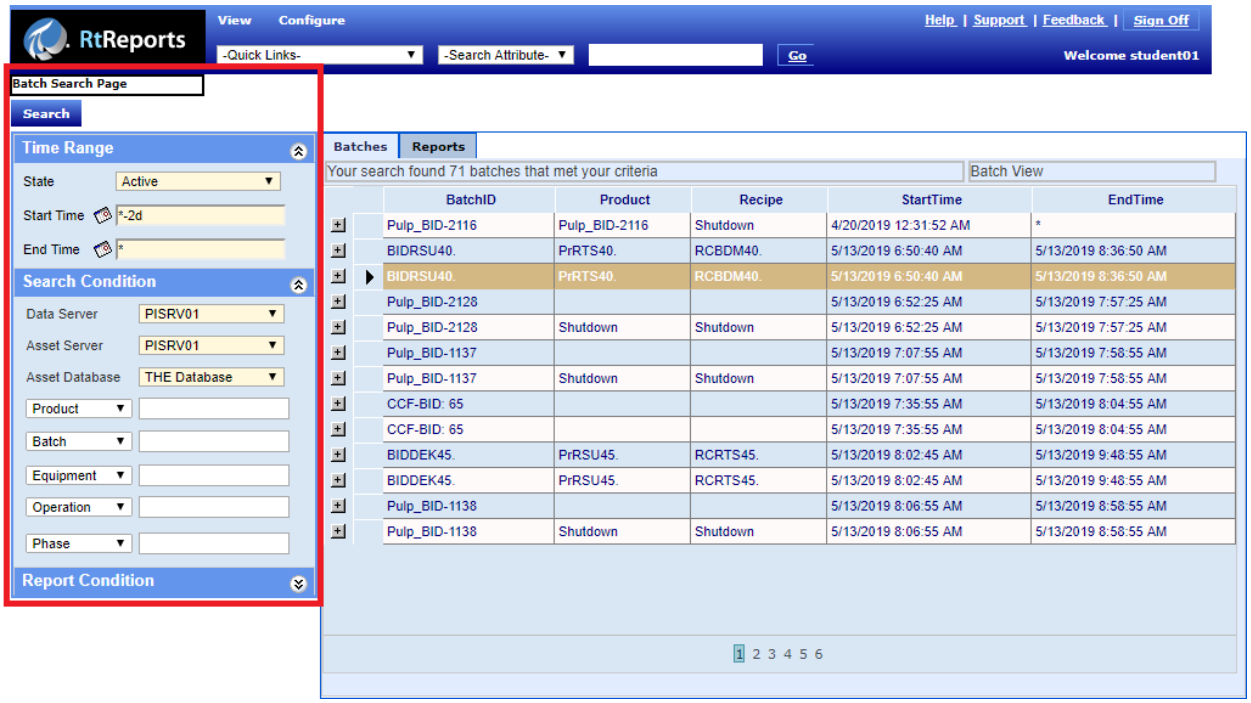
1. Launch RtReports Editor on the PIRPT01 Training Cloud Environment and Login
2. Open the *BatchRecords* report template -> Format Template tab -> select each section -> verify that the *Allow Verifications* and *Allow Comments* options are enabled (checked) -> OK -> File -> Save. Note: both options are checked by default



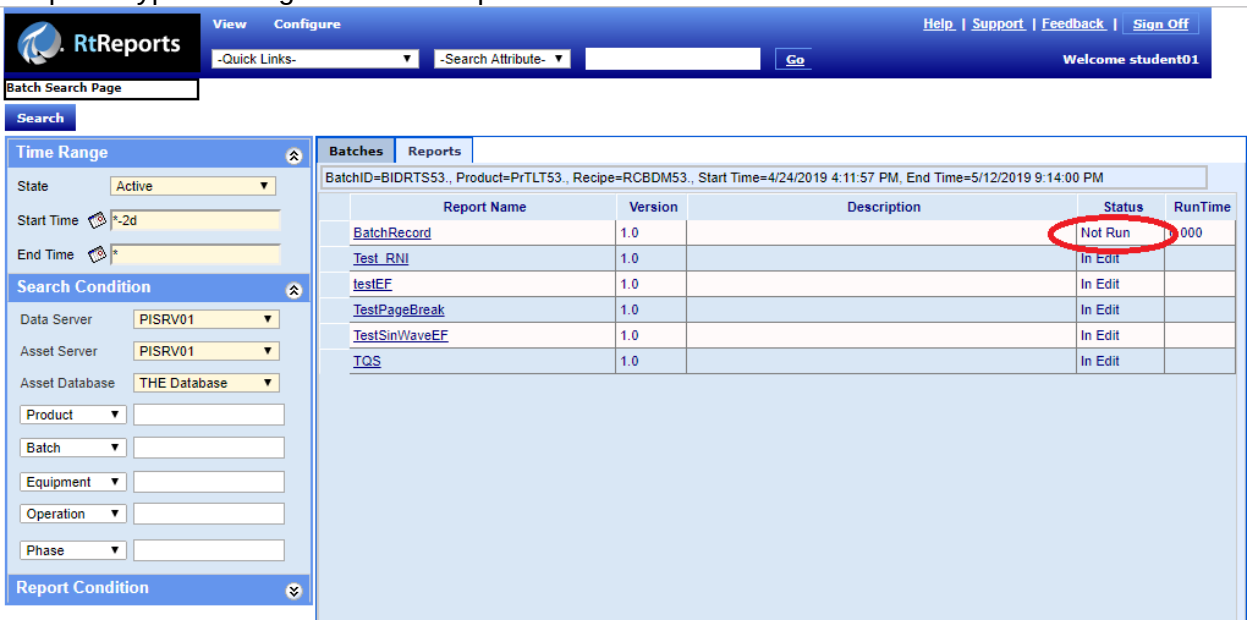
3. Release the report: Tools -> Report Version Manager… -> Review -> …
4. Launch RtReports Generator (on the PIRPT01 Training Cloud Environment) and login



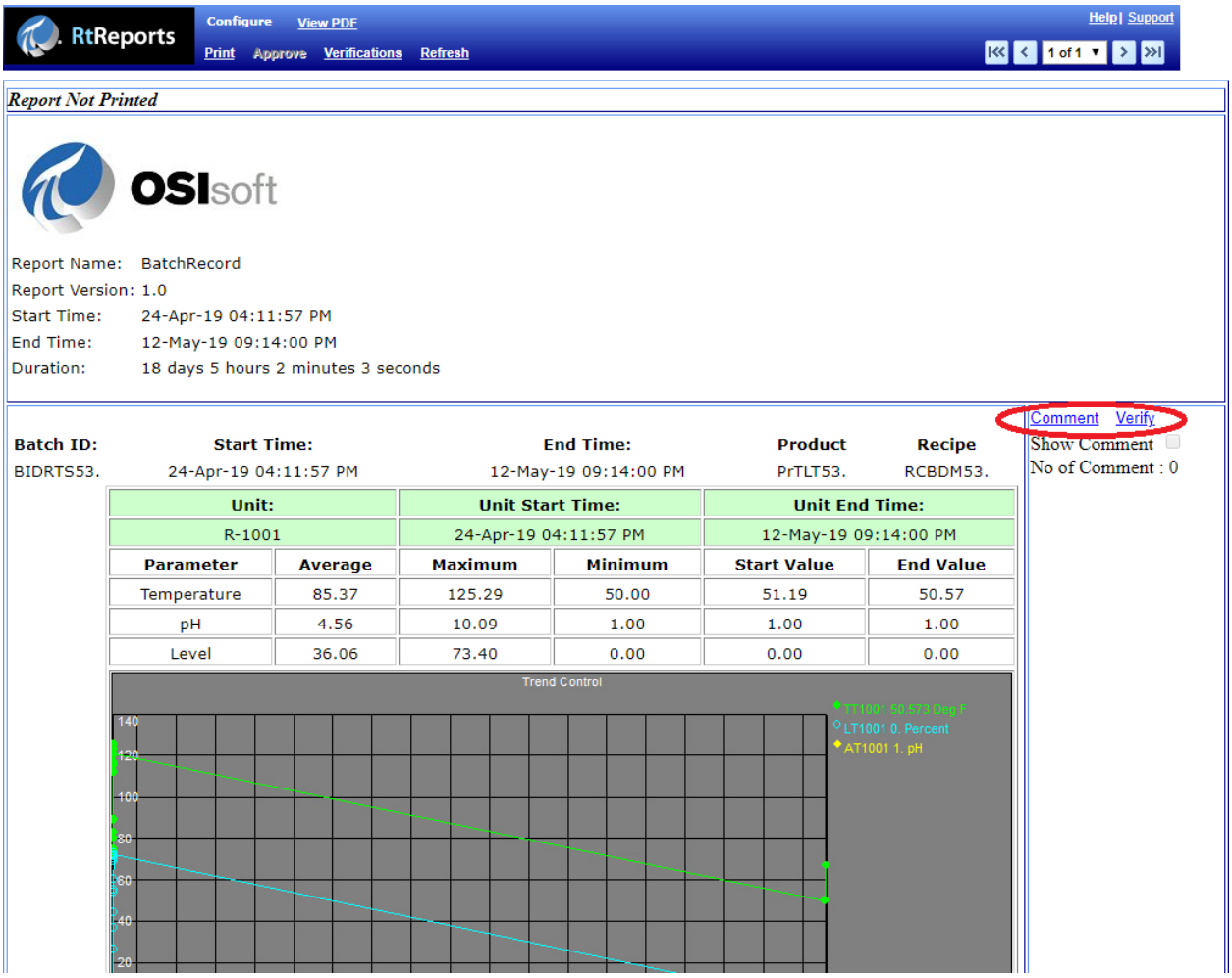
5. On the Batch Search Page, edit the Time Range, i.e. State, Start Time, and End Time -> edit the Search Condition, i.e. Data Server: PISRV01, Asset Server: PISRV01, and Asset Database: THE Database, also Product, Batch… of desired -> click Search

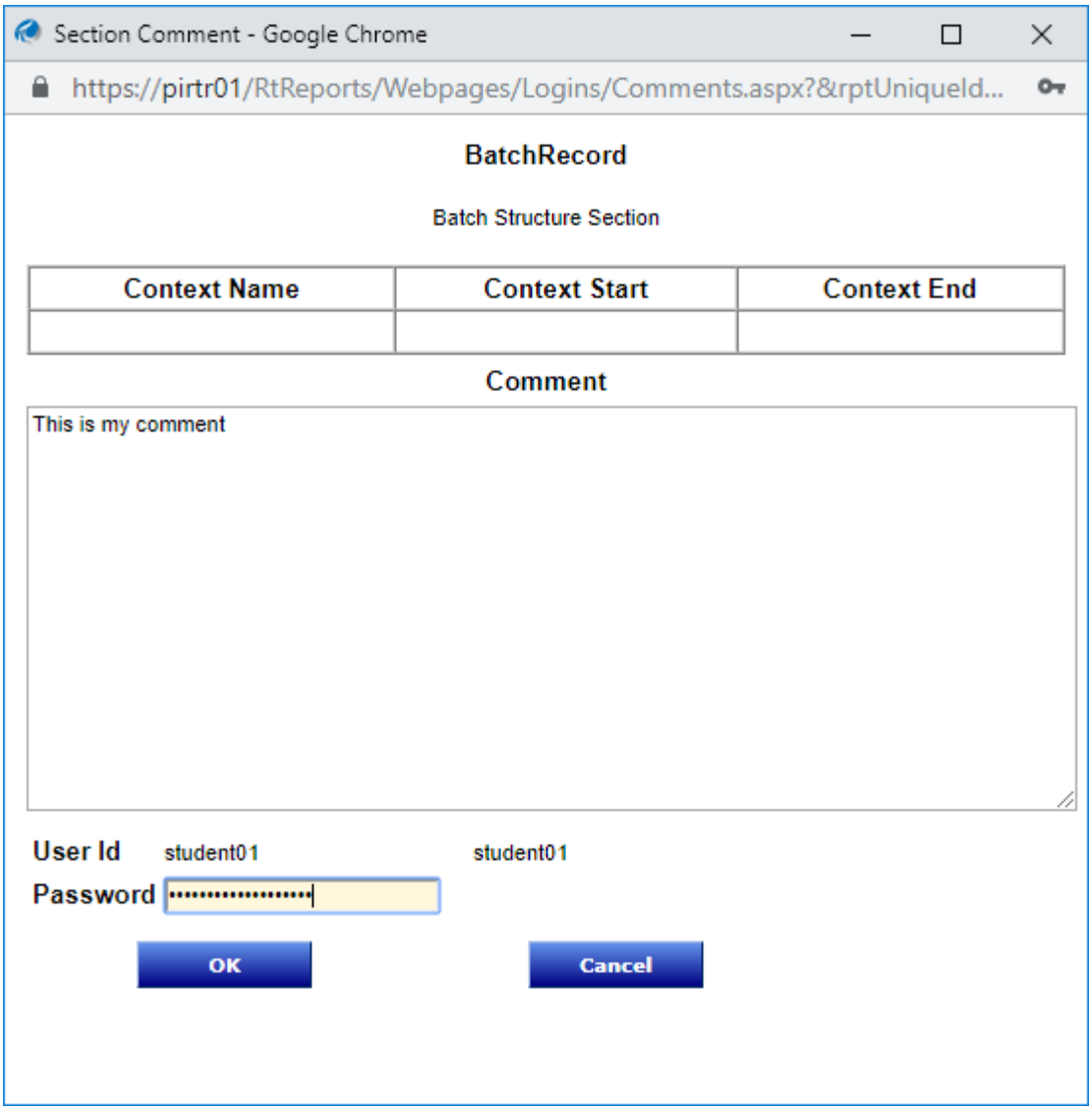


6. Select a completed BIDxxx batch (batch with start and end times) -> click Reports tab -> find and notice the BatchRecords Status: **Not Run** -> click the *BatchRecords* Batch report template hyperlink to generate the report

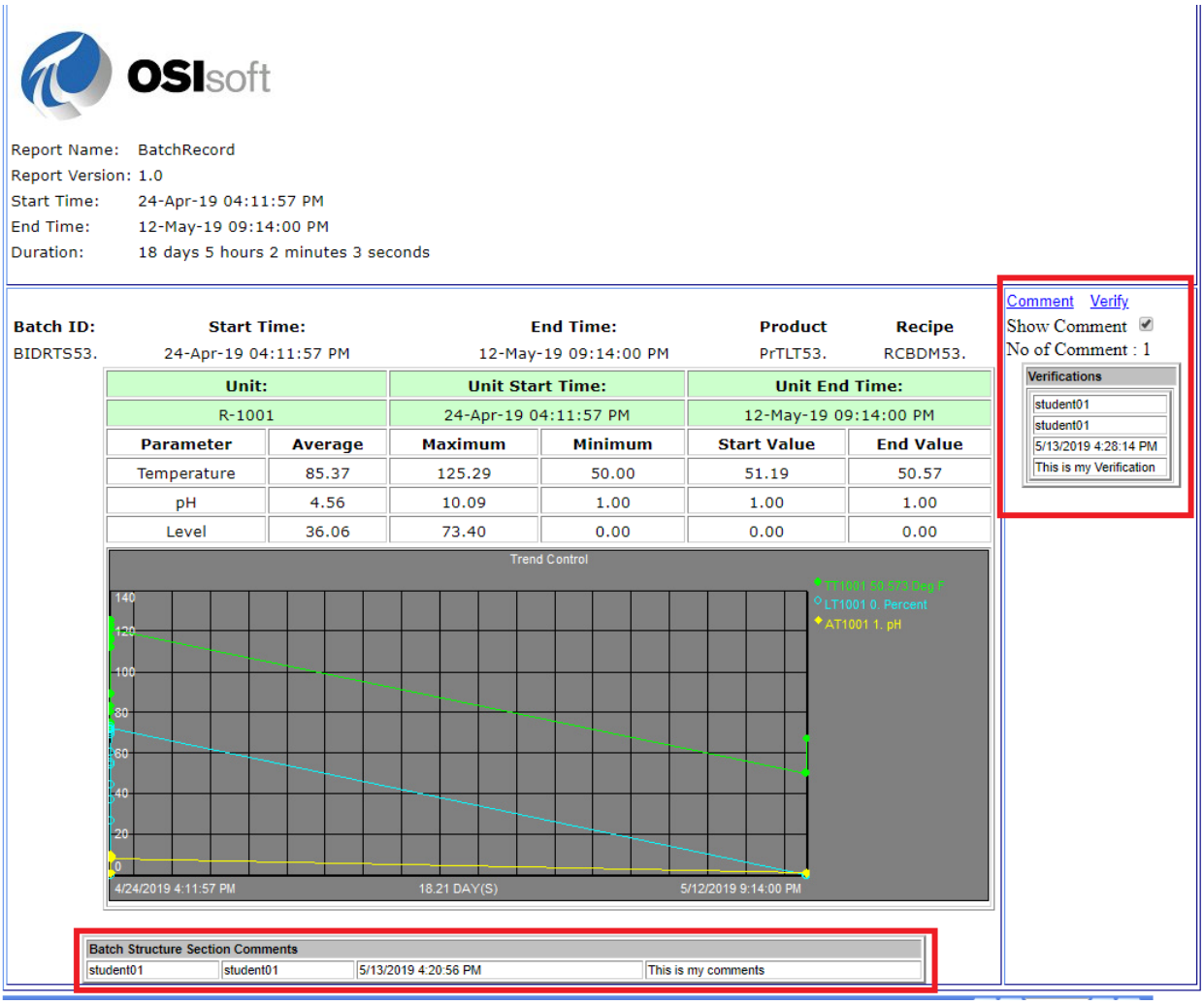


7. Wait for the report to appear, once the report is generated, click the Comment hyperlink -> enter comments -> enter Password -> OK

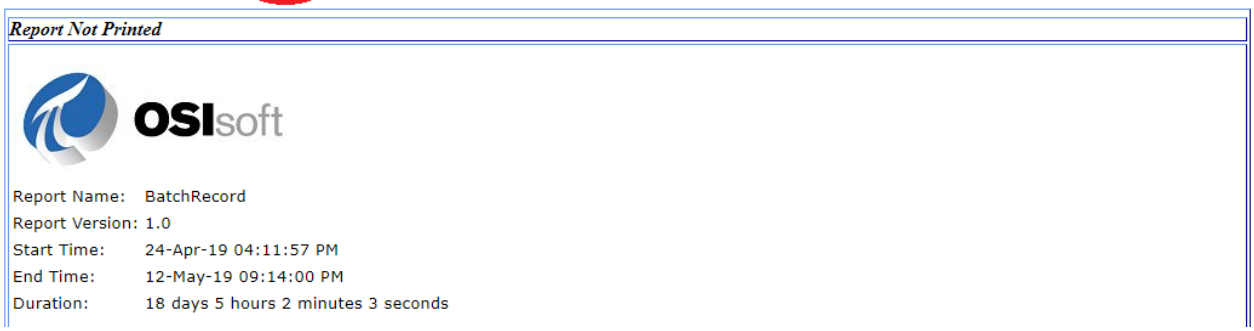

RtReports
Configure
View PDF
Print
Approve
Verifications
Refresh
Help| Support
1 of 1
Report Not Printed
OSIsoft
Report Name: BatchRecord
Report Version: 1.0
Start Time: 24-Apr-19 04:11:57 PM
End Time: 12-May-19 09:14:00 PM
Duration: 18 days 5 hours 2 minutes 3 seconds
Comment
Verify
Show Comment
No of Comment : 0
Batch ID: BIDRTS53.
Start Time: 24-Apr-19 04:11:57 PM
End Time: 12-May-19 09:14:00 PM
Product: PrTLT53.
Recipe: RCBDM53.
Unit: R-1001
Unit Start Time: 24-Apr-19 04:11:57 PM
Unit End Time: 12-May-19 09:14:00 PM
Parameter | Average | Maximum | Minimum | Start Value | End Value
Temperature | 85.37 | 125.29 | 50.00 | 51.19 | 50.57
pH | 4.56 | 10.09 | 1.00 | 1.00 | 1.00
Level | 36.06 | 73.40 | 0.00 | 0.00 | 0.00
Trend Control
LT1001 0. Percent
AT1001 1. pH
140
120
100
80
60
40
20



8. Notice *Show Comment* is checked and Batch Structure Section *Comments* (at the bottom)
9. Click *Verify* hyperlink -> repeat the same steps to enter *Verification(s)*



10. Notice the Verifications…
11. Also note that the *Approve* hyperlink is disable (gray out); that is because the report template was configured without Approval cycle



Exercise 14: Generating a Time-Based Report

Objectives

Given a Time-Based Template, generate and view multiple reports.

Problem Description

Generate and view multiple daily time reports using the *DailyReport* time-based template imported in earlier exercise.

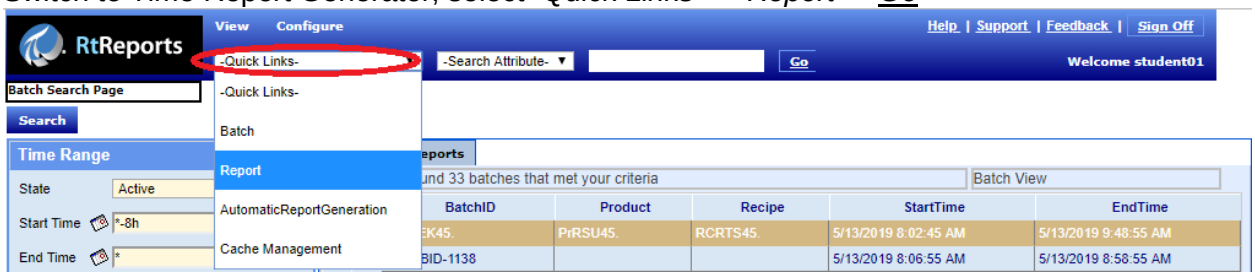
Approach

Using the RtReports Generator, choose the time-based report search and select the *DailyReport*.

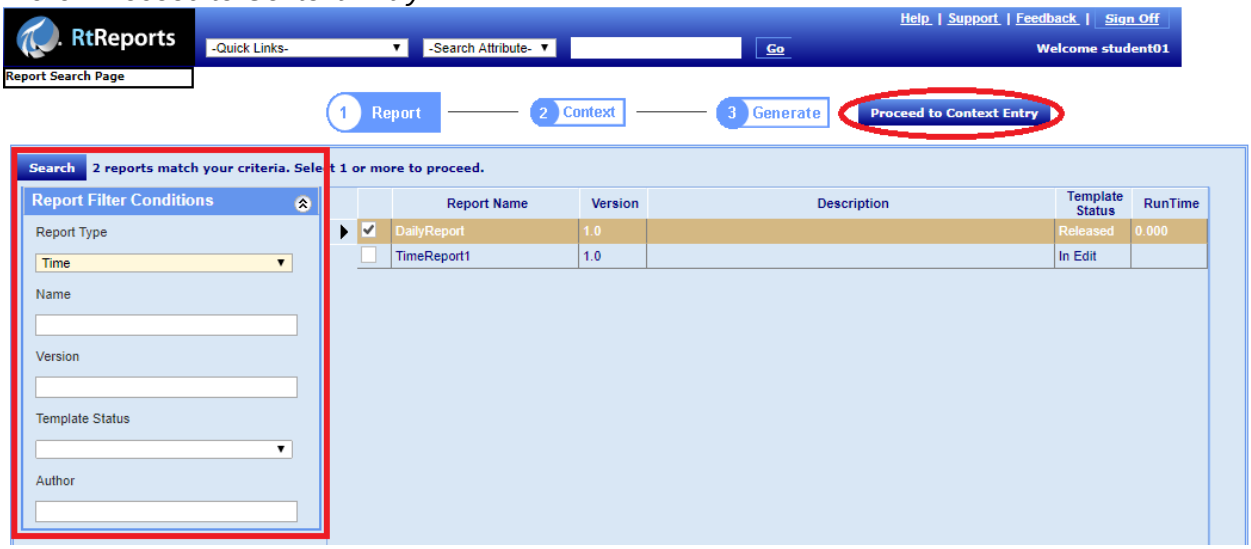
Proceed to the context search and search for the last 7 days.

Solution 14: Generating Multiple Time-Based Reports

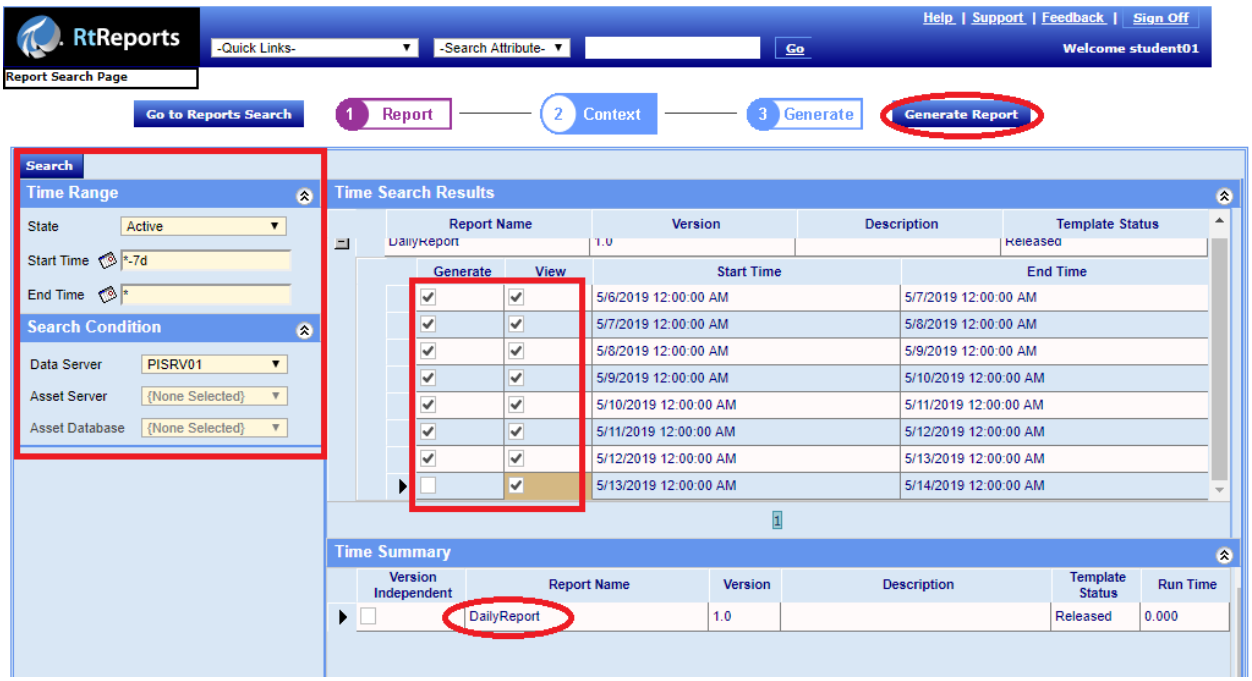
1. Launch RtReport Editor on the PIRPT01 Training Cloud Environment and login
2. Open the *DailyReport* from previous exercise -> verify each section *Allow Verifications* and *Allow Comments* options are enabled (checked) -> Save. Note: both options are enabled by default
3. Release the DailyReport, Tools -> Report Version Manager…
4. Launch RtReports Generator and login
5. Switch to Time Report Generator, select *-Quick Links- -> Report -> Go*



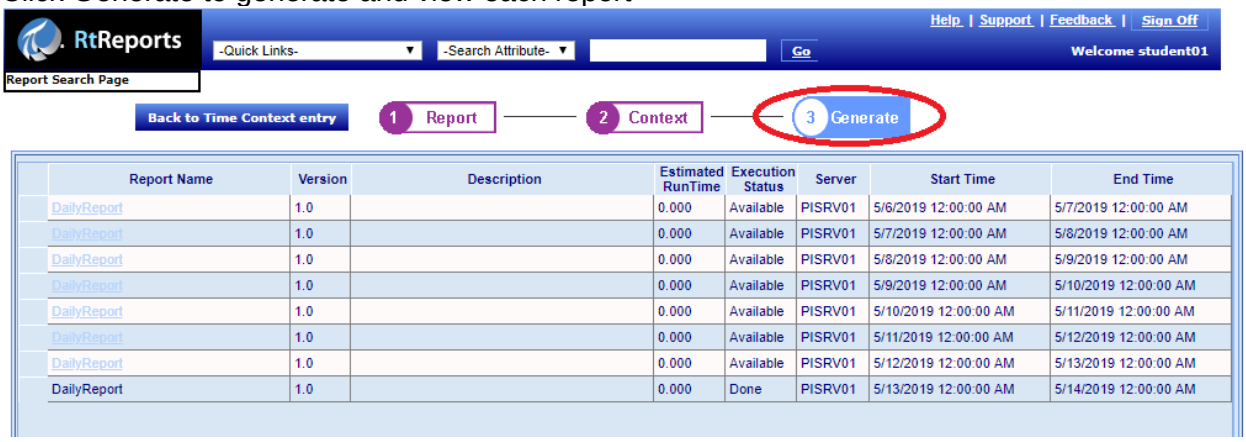
Fill in the *Report Filter Conditions* -> click *Search* -> select the *DailyReport* report template -> click *Proceed to Context Entry*



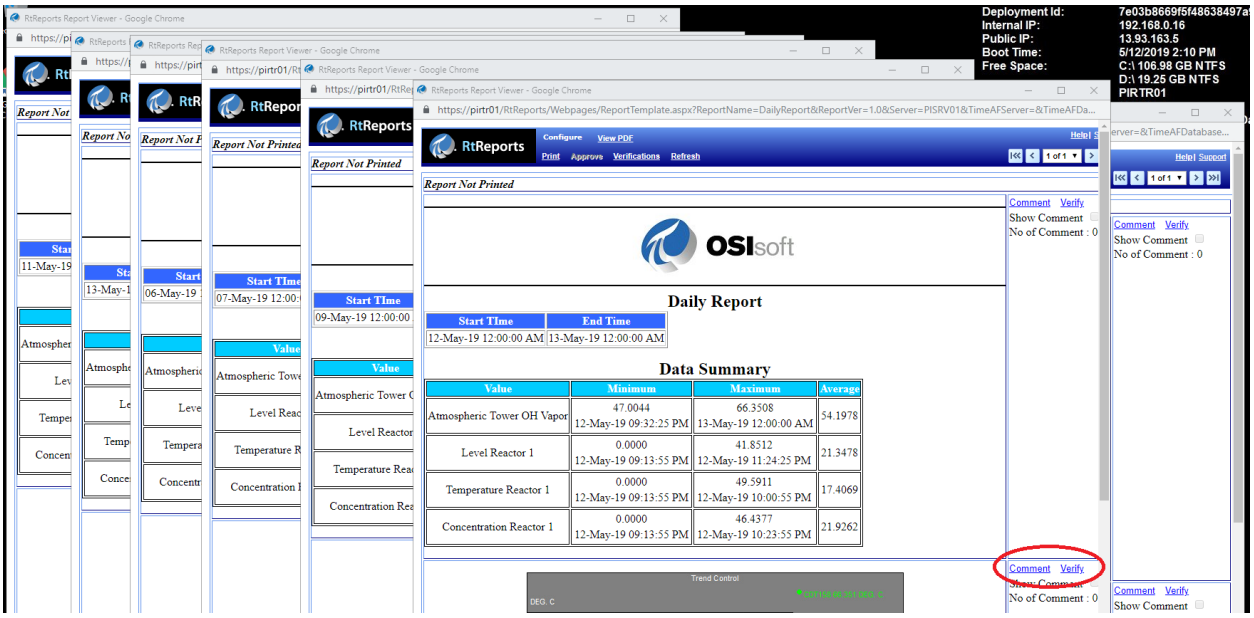
6. Enter Start Time: *-7d and End Time: * -> Data Server: PISRV01 (note: Asset Server and Asset Database are required when report template is using AF attributes) -> click *Search* again -> expand DailyReport -> check *Generate* (to generate a report, report templated must be released and end time must be in the past) and *View* for each report -> click *Generate Report*. **Notice,** each selected report is generated (one report per html page)



7. Click Generate to generate and view each report



8. Select each report (one at a time) -> select *Comment* hyperlink to enter comments -> and select *Verify* hyperlink to enter verifications


Deployment Id:
Internal IP:
Public IP:
Boot Time:
Free Space:
192.168.0.16
13.93.163.5
5/12/2019 2:10 PM
C:\ 106.98 GB NTFS
D:\ 19.25 GB NTFS
PIRTR01
RtReports
Configure
View PDF
Print
Approve
Verifications
Refresh
Report Not Printed
OSIsoft
Daily Report
Start TIme
End Time
12-May-19 12:00:00 AM
13-May-19 12:00:00 AM
Data Summary
Value
Minimum
Maximum
Average
Atmospheric Tower OH Vapor
47.0044
12-May-19 09:32:25 PM
66.3508
13-May-19 12:00:00 AM
54.1978
Level Reactor 1
0.0000
12-May-19 09:13:55 PM
41.8512
12-May-19 11:24:25 PM
21.3478
Temperature Reactor 1
0.0000
12-May-19 09:13:55 PM
49.5911
12-May-19 10:00:55 PM
17.4069
Concentration Reactor 1
0.0000
12-May-19 09:13:55 PM
46.4377
12-May-19 10:23:55 PM
21.9262
Comment
Verify
Show Comment
No of Comment : 0
Trend Control
DEG. C

Run Time Action Parameters

Beginning with RtReports version 3.2, template designers can create report templates with report parameters which can be changed at runtime in the RtReports Generator. This allows for powerful ad hoc reporting.

Implementing Run Time Parameters

Parameters are designed in the Data Template Root. You must define all run time parameters in the Data Template before they are used in a report. In addition to the parameter name, you will define the text string the user sees in the Generator and the behavior.

If you do not define a valid default value the report will fail on testing.

Once the parameters are created, they can be used in the report. These report parameters will be used in the logic of the report – and that usually means a Journal Action. To use the parameter in a Journal Action, surround the parameter name with square brackets.

For example, you can define a parameter called usertagname. Then you will create an action parameter of a type PI Tag and use the Value [usertagname].

Exercise 15: Run Time Parameters for Batch Report

Objectives

Create a report that allows users to select parameter(s).

Problem Description

Configure a Batch report that allows users to report on their picks of Unit (AF element) and AF Attribute from lists.

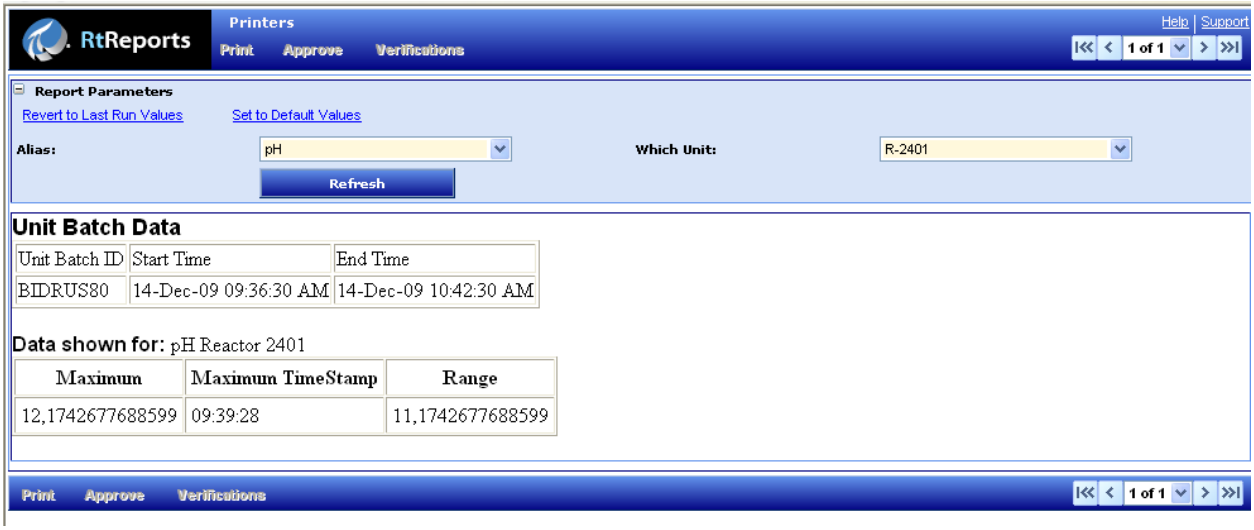
Approach

- Create a new Batch Report “Run Time Parameters Report”.
- Configure “Units” Run Time Parameter of type List that consists of R-1001 and R-1002.
- Configure “Attributes” Run Time Parameter of type List that consists of Level, pH, Temp.
- Syntax for using a Run Time Parameters: [*Run Time Parameters Name*]
- Insert a Unit Journal Action “SummaryJournalAction” of Result Definitions type “Summary Value” of Minimum, Maximum, Average, and Range of the selected Unit\Attribute.
- Display the Result Definitions of the SummaryJournalAction using the Trace Set object

Hint: Watch for absolute references to Unit (filter pattern: [Name:”R-1001”]) in the source xml path.

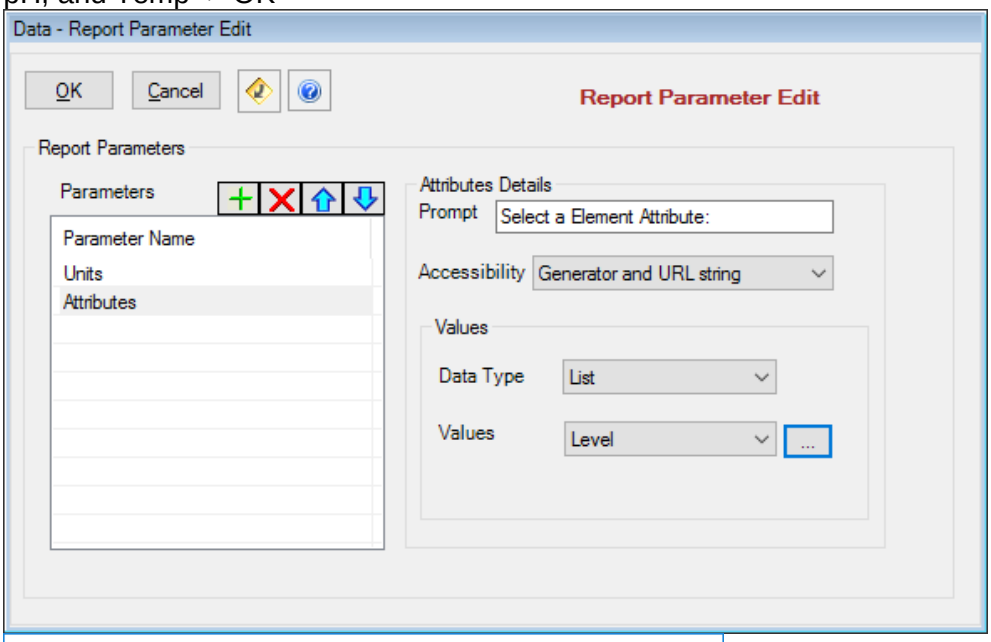
Test the report against an EF of the 1K Reactors by selecting a Units and/or an Attributes and Refresh.

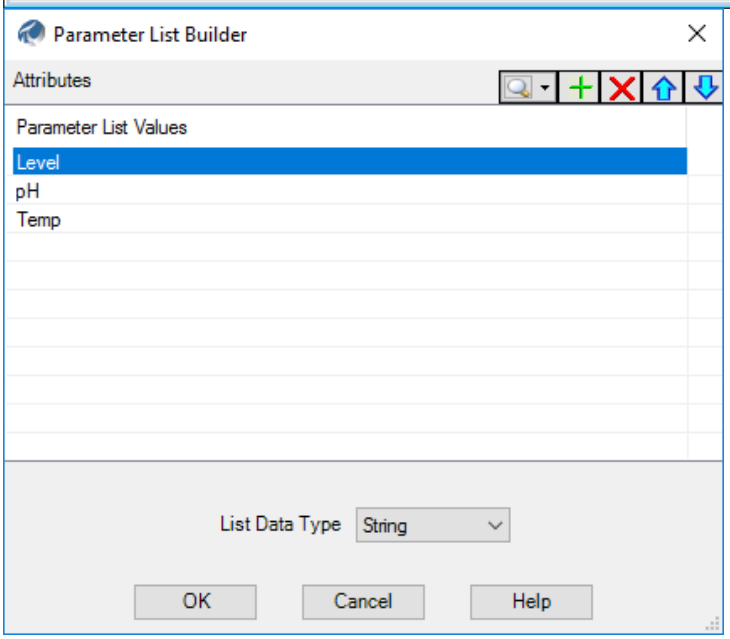
The report should look like the following:



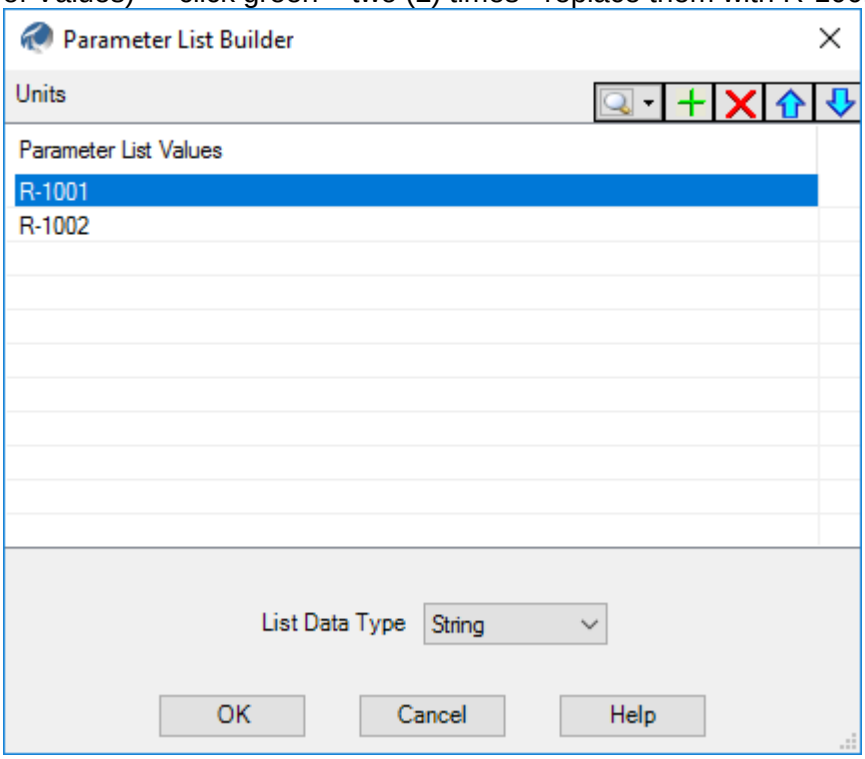
Solution 15: Run Time Parameters for Batch Report

1. Launch the RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Configure new Batch Report *"Run Time Parameters Batch Report":* File > New > Batch Report Template...
3. Add the *"Attributes"* Run Time Parameters: Tools -> Report Parameters -> green + -> call it "Attributes" -> Alias Details Prompt: "Select an Element Attribute: " -> Data Type = List -> click … (to the right of Values) -> click green + three (3) times> replace them with Level, pH, and Temp -> OK

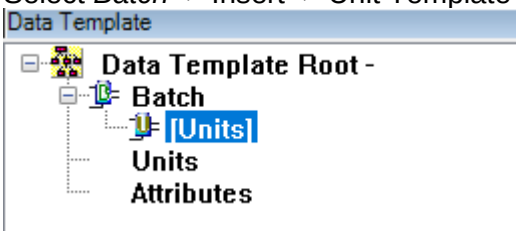




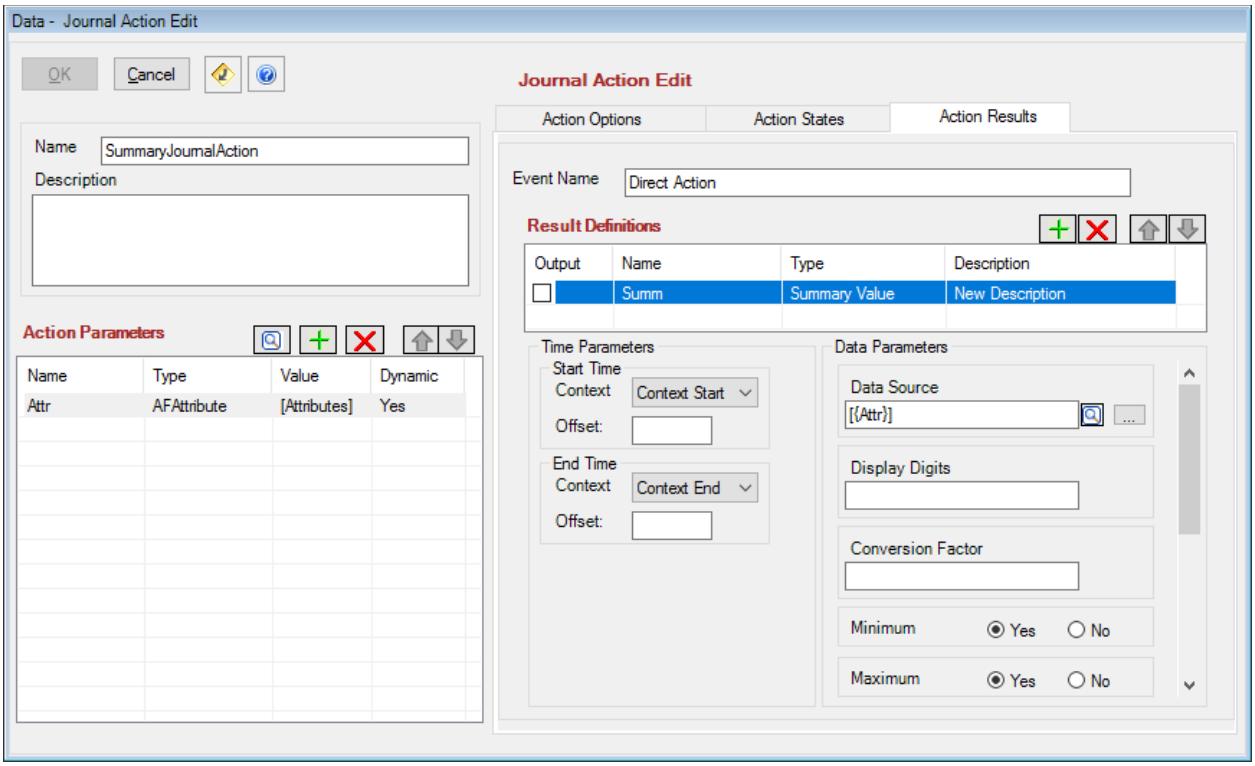
4. Add the *"Units"* Run Time Parameters: Tools > Report Parameters -> green + -> call it "Units" -> Alias Details Prompt: "Select a Unit: " -> Data Type = List -> click … (to the right of Values) -> click green + two (2) times> replace them with R-1001 and R-1002 -> OK



5. Click OK to complete the Run Time Parameters configuration
6. Switch to the Data Template tab
7. Select *Batch* -> Insert -> Unit Template -> Name: [Units] -> OK



8. Select the [Units] Unit Template -> Insert -> Journal Action -> Name: SummaryJournalAction…
9. Action Parameters: green + -> Name: Attr -> Type: AFAttribute -> Cancel the Attribute Search prompt -> Value: [Attributes]
10. Journal Action Edit: Action Results tab
11. Result Definitions: green + -> Name: Summ -> Type: Summary Value -> drag Attr to Data Source -> check Yes for Minimum, Maximum, and Average -> OK

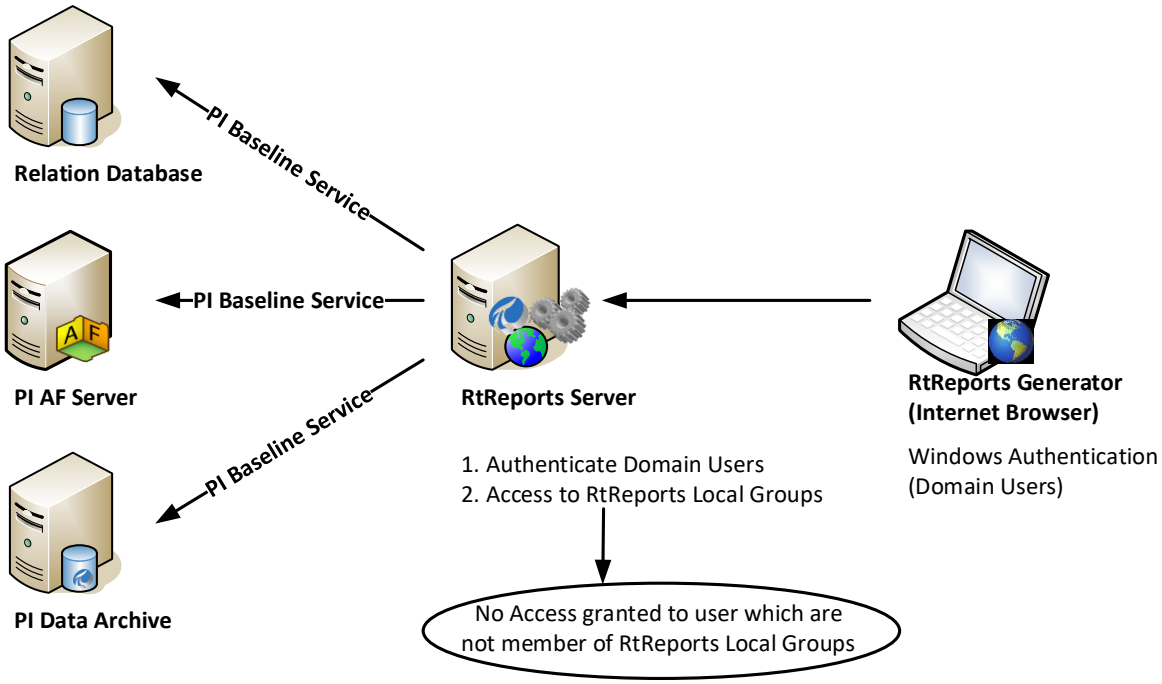


12. Save Report Template -> View -> Test Report -> Use Event Frames (or F3) -> search a completed Procedure of 1K Reactors…
13. Switch to the Format Template tab
14. Select root -> Insert Section
15. Expand Section -> select Entry -> Inset -> Trace Set -> browse and drag *BatchJournal/UnitBatchJournals/UnitBatchJournal[Name="R-1001"]/JournalEvents/JournalEvent[Name="SummaryJournalAction"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Summ"]/Value/TraceSet[@Name="Summ.Summary"]* to Source field -> Border tab -> Frame: Border -> Border Width: 1 -> Rows Definition tab -> red X columns Time, ErrorCode, ErrorDescription, and IsGood -> OK
16. Save Report Template -> Refresh (of F3)

Lesson 10: PI RtReports Security

Introduction

The PI RtReports uses Windows Authentication. If user login successfully, the system checks to see what groups include the user on the PI RtReports Server. This group list is then checked against the PI RtReports Security database to make sure the user belongs to a valid RtReports group, which means PI RtReports security attributes have been granted to the group. If the user has the basic security attribute to log into PI RtReports, the login credentials are then passed to the PI System for authentication against the Trust Table. If this validation is again successful, the user is granted access to the PI RtReports application.



A user who has access to the RtReports application may use the Generator and depending on attributes, may use the Editor. However, the user must also be granted access to edit or run individual Report Templates from within the Editor

RtReports Windows Groups

RtReports uses local Windows groups created locally to grant privileges to the users. For each report, the author can assign which group or groups who will have the

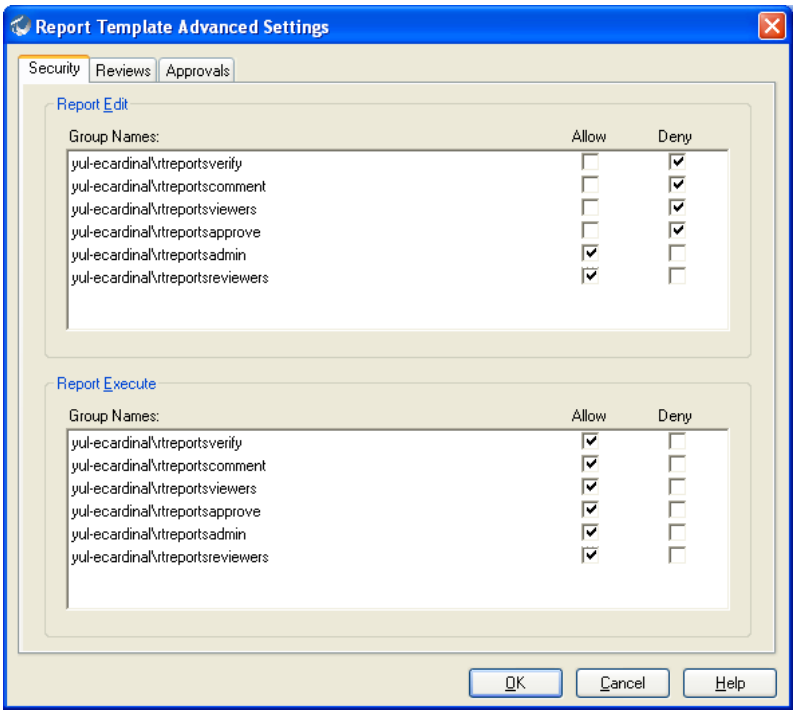
Editing, Administrating, Reviewing, Printing, Officially Printing, Commenting, Verifying and Approving the reports. The users requiring privileges will be added to the proper group.

More than one group might be required for each task. For example, if you have two sections (A and B) and you have a report for A and one for B. The approvers for A and B may not be the same persons necessarily. In this case, you might need two groups RtReportsApproversA and RtReportsApproversB.

The management of the local group is done using the Microsoft Computer Management Console.

Template Security

Report Edit Security refers to the permission to edit a specific Report Template. Groups that are listed in the Report Edit list and given the "Allow" permission, mean member(s) of the groups may open the specific Report Template for editing. Group(s) with "Deny" permission, mean member(s) of the groups may open and view the Report Template (editing is "Deny").



Security is individually set for each template. Report Edit lists the groups that can edit a report. Report Execute lists the groups that can run a report.

Report Execute Security

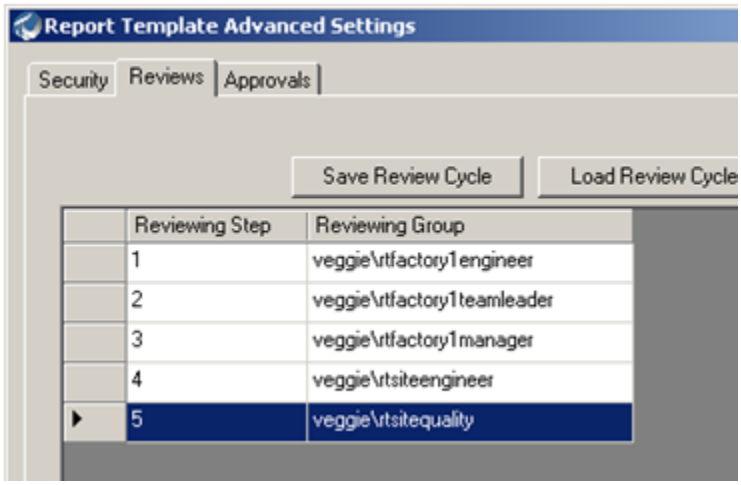
Report Execute Security refers to the permission to execute a specific Report Template from the Generator. If a group is listed in the Report Execute list and given a permission of Allow, then any user within this group may run the report. If a group is listed in the Report Execute list and given a permission of Deny, then any user within this group will not see this Report Template in the RtReports Generator.

Report Template Review Cycle

Users have the option of adding one or multiple "Review Cycle" for each Report Template prior to be released. Review Cycle options:

- "NO Review Cycle" – report template Status change from "In Edit" -> "Released"
- One or multiple Review Cycle – report template Status change from "In Edit" -> "In Review Step 1" -> "In Review Step x"-> "Released". If rejected at any Review Cycle, the report template Status will switch back to "In Edit"

By configuring a Review Cycle, users can capture multiple electronic signatures to approve the new version of the Report Template.



Procedures to setup Review Cycle: RtReports Editor -> open the Report Template -> Format Template tab -> select/right click on the "root" (**Name** of the Report Template) -> Properties -> on the popup, click the Advance… button -> on the "Report Template Advanced Settings popup, select the **Reviews** tab -> click "+" to add "Reviewing Step" -> select "Reviewing Group" from the dropdown. **NOTE:** Only groups already granted the review security permission are displayed in the list.

To add the next (more) "Reviewing Step": start by clicking on the "+" -> …

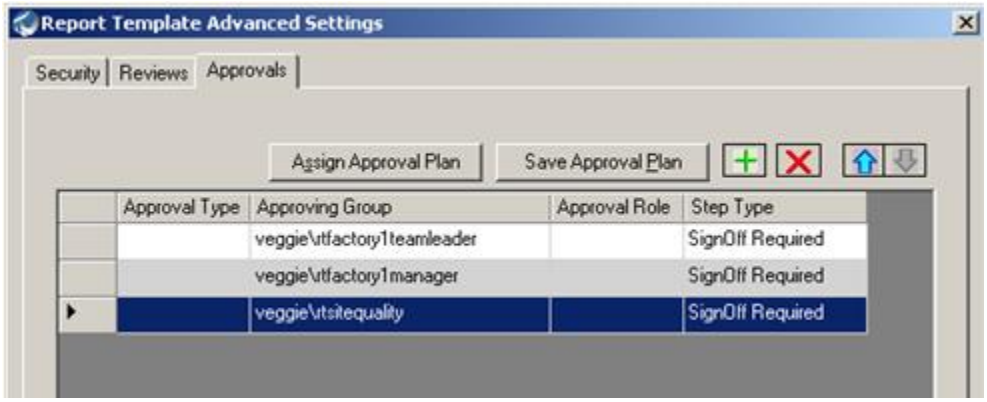
To re-order the Review Steps: select the Review Step -> use the up/down arrow…

To delete the Review Step: select the Reviewing Step -> click the "X"

Configure an Approval Plan for a Generated Report

Users may want to specify a Production requirement for approving report. Options for the Approval Plan:

- "NO Approval Plan" – report Status change from "Not Run" -> "Not Yet Approve" -> "Approved"
- "One or multiple Approval Plan" – report Status change from "Not Run" -> "Not Yet Approve" -> "**In Approval Cycle 2"** -> "In Approval Cycle X" -> Approved"



Procedures to setup Approval Cycle: RtReports Editor -> open the Report Template -> Format Template tab -> select/right click on the "root" (**Name** of the Report Template) -> Properties -> on the popup, click the Advance… button -> on the "Report Template Advanced Settings popup, select the **Approvals** tab -> click "+" to add "Approval Type" -> specify the Approval Type -> select "Reviewing Group" from the dropdown (**NOTE:** Only groups already granted the approve security permission are displayed in the list) -> specify the Approval Role -> Step Type "SignOff Required"

To add the next (more) "Approval Cycle": start by clicking on the "+" -> …

To re-order the Approval Cycle: select the Approval Type -> use the up/down arrow…

To delete the Approval Cycle: select the Approval Type -> click the "X"

Exercise 16: Group and User Security

Objectives

Given users and groups below; setup desired RtReports security.

Problem Description

Given the list of usernames and groups below; setup the security parameters in order to match the current description.

Windows Group	Domain User	Permission
RtReportsApprovers	LBeethoven	All permission except Administration
RtReportsCommenters	LBernstein	User, Comments, and Print
RtReportsReviewers	ACLBizet	User, Edit, Review, and Print
RtReportsUsers	GFFVerdi	User and Print
RtReportsVerifiers	WAMozart	User, Verifications, and Print

Note: You need to be an Administrator of the RtReports Server in order to create the groups

User Password: P1school! (for all users).

Approach

Use the Computer Management Console to create the Windows groups locally on the RtReports server and add the users.

Use the RtReports Group Privileges tool to setup the access rights based on the Windows groups.

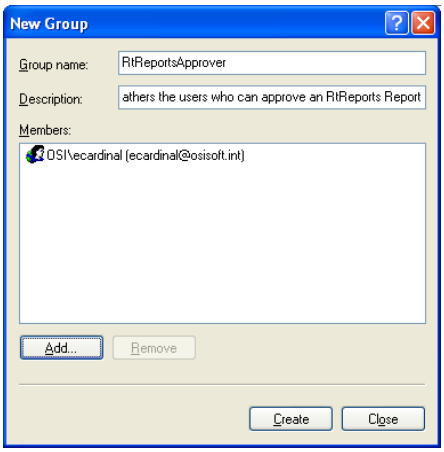
Open your report and setup the security settings from the “Advanced” button on the Root Format Template property window. Setup the Review and Approval Cycles.

Test the application with different users.

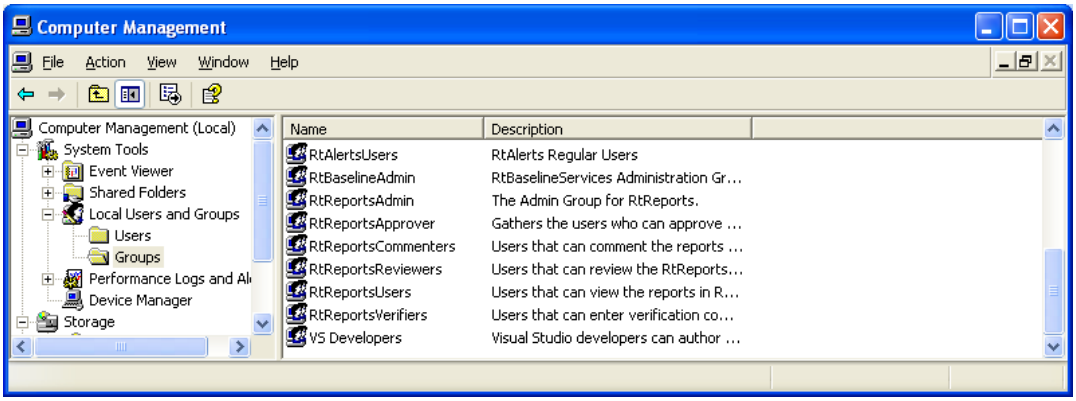
Solution 16: Setup Group and Users Security

Create the Local Windows Groups

1. Launch the Microsoft Computer Console Management (on the PIRPT01 Training Cloud Environment): Start -> Control Panel -> Administrative Tools -> Computer Management.
2. Expand the Select the Local Users and Groups and choose the Groups folder.
3. Right-click and select the New Group… menu.
4. Enter RtReportsApprovers as the Group Name.
5. Click on the Add button to include the “LBeethoven” user.
6. Click on Create.

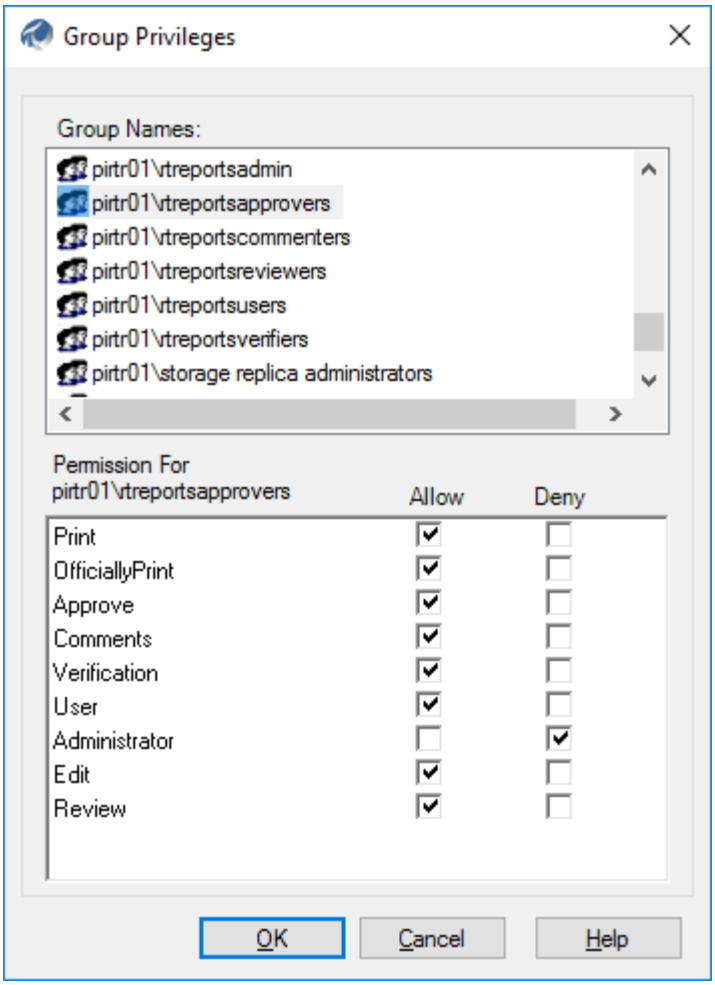


7. Repeat steps 3 to 6 for all the other groups.



Setup Group Privileges in RtReports

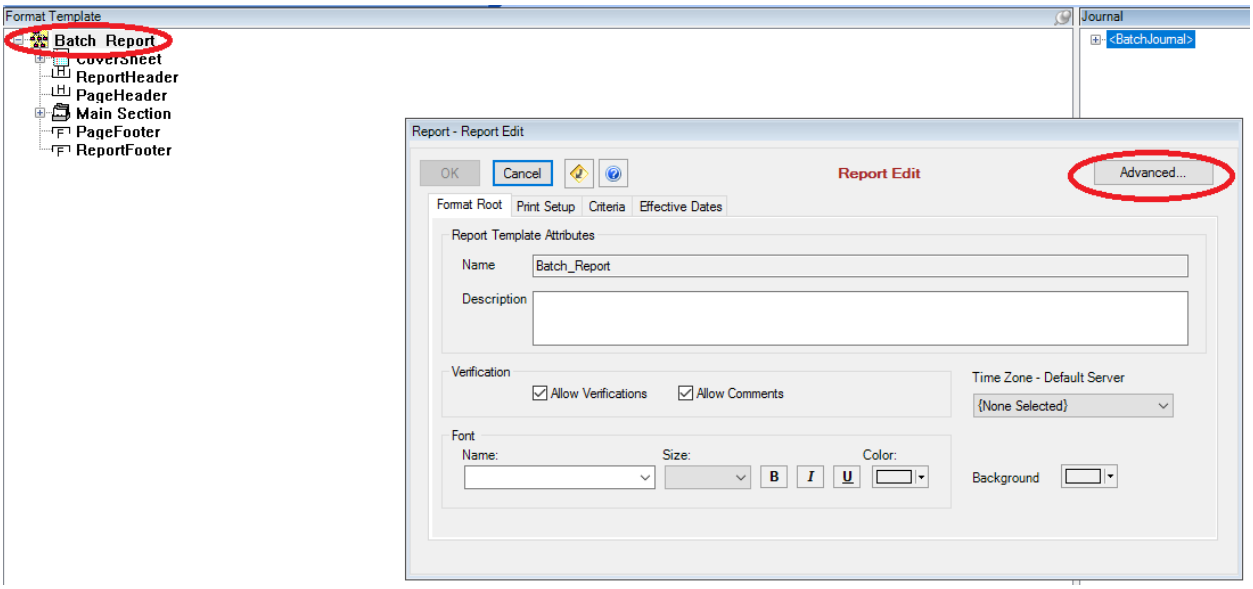
1. Launch RtReports Editor on the PIRPT01 Training Cloud Environment and Login
2. Admin menu -> Group Privileges…
3. Under *Group Names:,* browse the local group list -> select the RtReportsApprovers -> in the *Permission for <Domain>\<Group>* allow all permissions except the Administrator



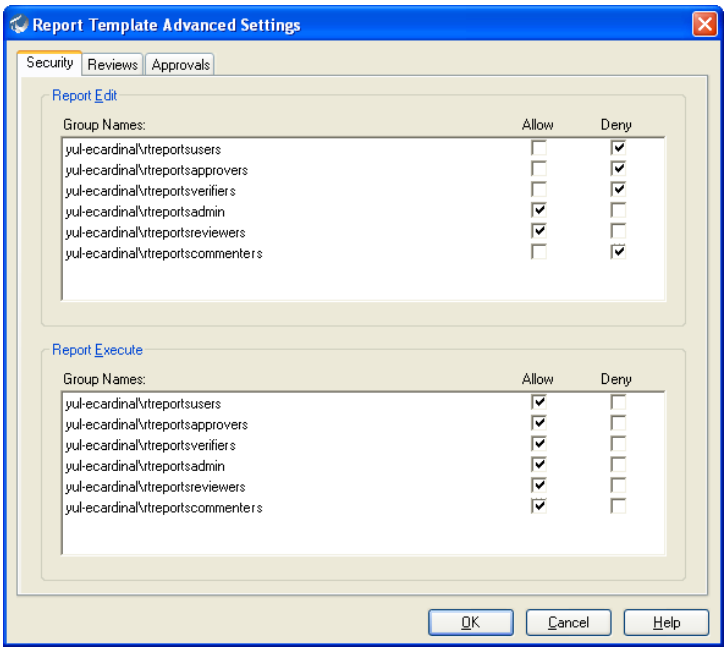
4. Repeat step 3 for the followings:
   - RtReportsCommenters group and allow Comments, User, and Print permissions
   - RtReportsReviewers group and allow Edit, Review, User, and Print permissions
   - RtReportsUsers group and allow Print and User permissions
   - RtReportsVerifiers group and allow Verification, User, and Print permissions

Setup individual Report Security

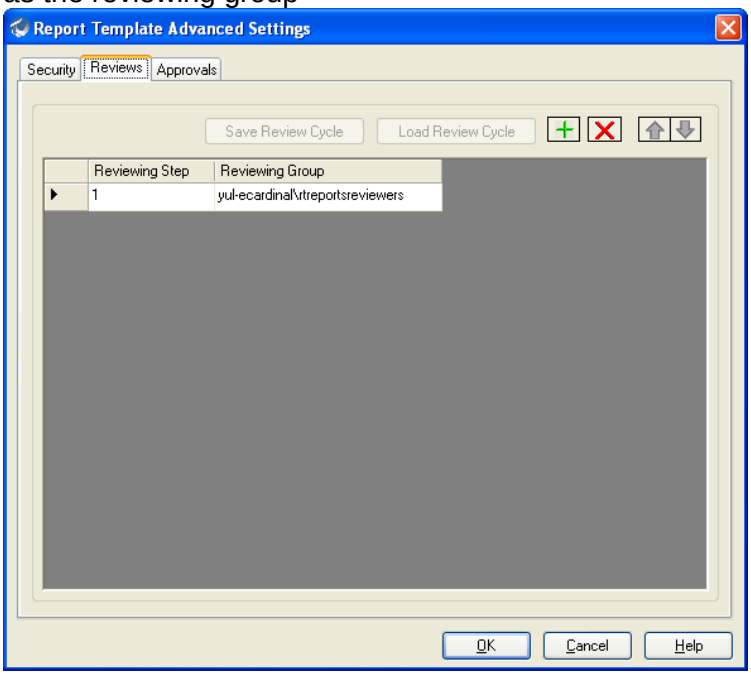
1. Launch RtReports Editor on the PIRPT01 Training Cloud Environment and login
2. Open the Batch_Report report template -> Format Template -> double clicks the report *root* -> Advanced… ->



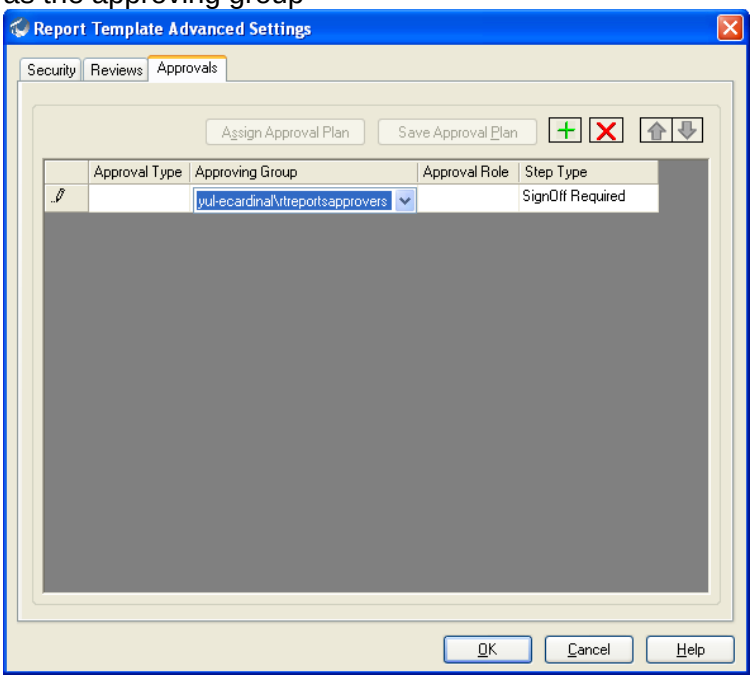
3. Select Security tab ->
   - In Report Edit -> Group Names: check *Allow* for local groups RtReportsAdmin and RtReportsReviewers
   - In Report Execute -> Group Names: check *Allow* all for local groups RtReportsAdmin, RtReportsApprovers, RtReportsCommenters, RtReportsReviewers, RtReportsUsers, and RtReportsVerifiers
   - OK



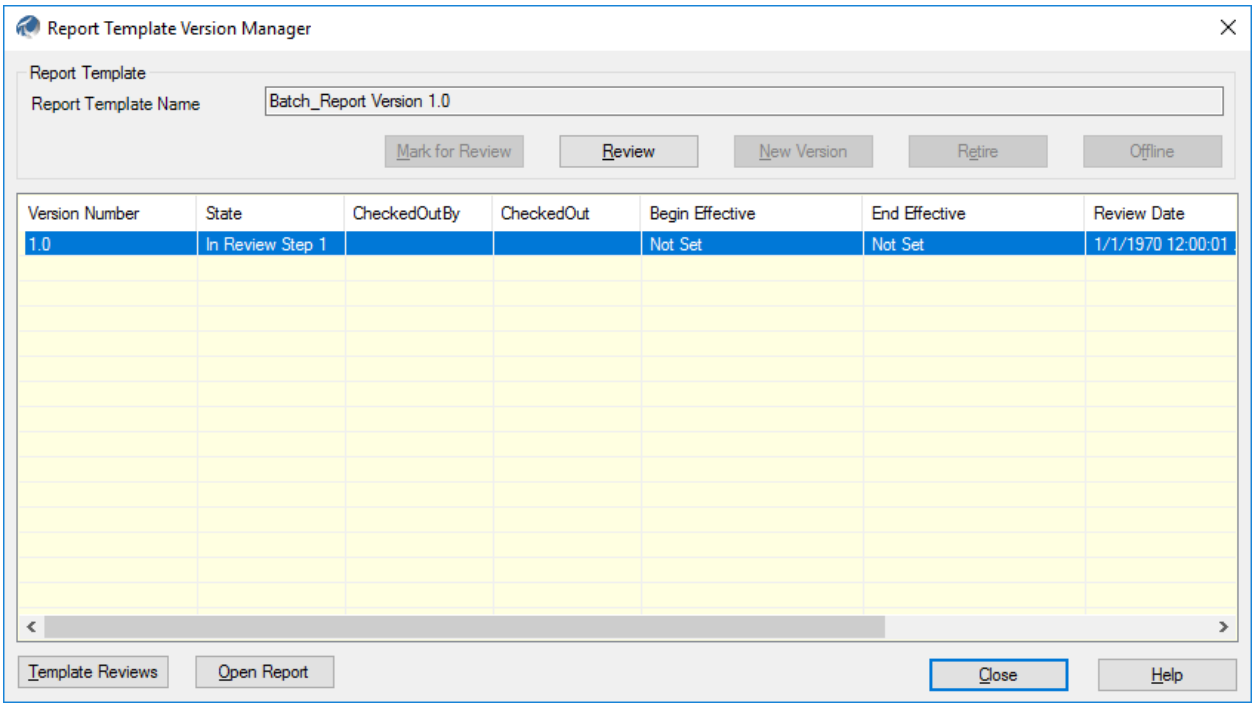
1. Select Reviews tab -> click the green + button -> browse and select RtReportsReviewers as the reviewing group



2. Select Approvals tab -> click the green + button -> browse and select RtReportsApprovers as the approving group



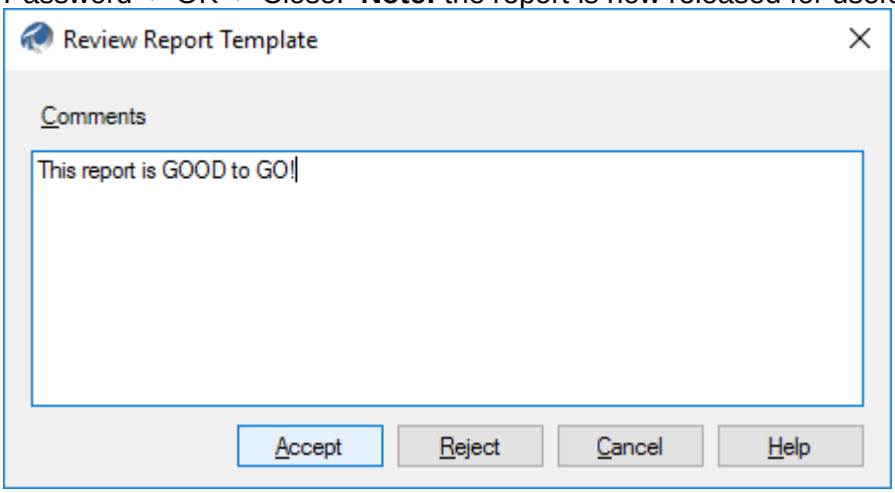
3. Click OK -> OK
4. To begin reviewing the report: Tools -> Report Version Manager -> click Mark for Review -> Close -> Save the report. **Note:** the report is ready for Review and Release



5. Exit Report Editor

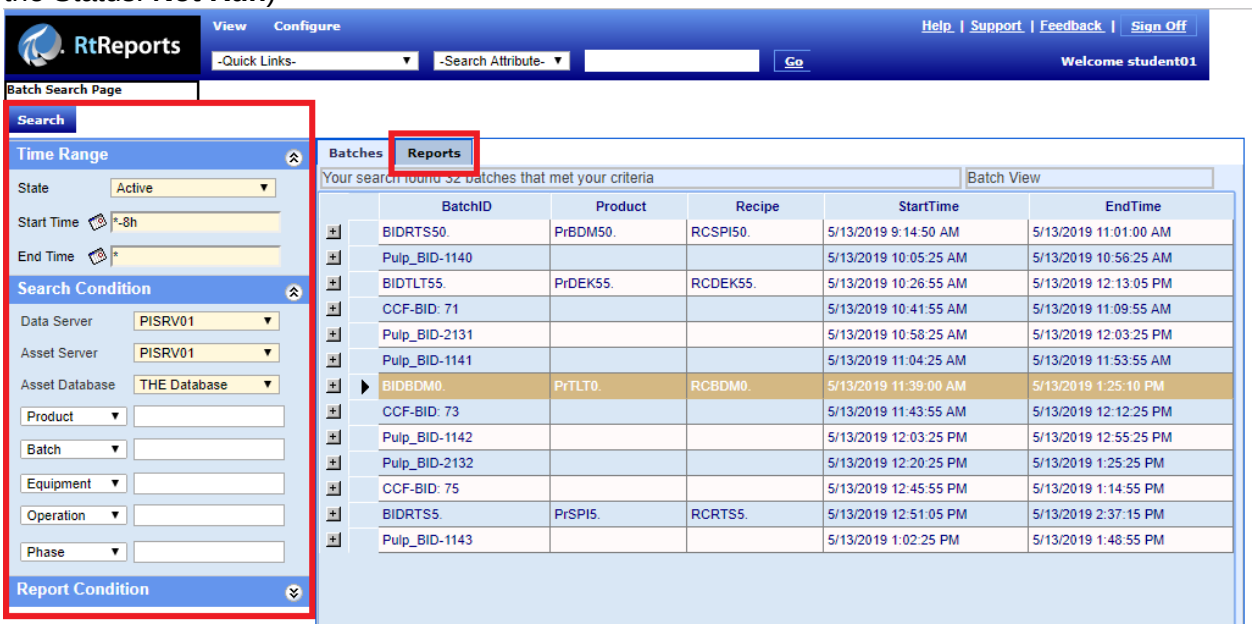
Review and Release report

1. Launch RtReports Editor on the PIRPT01 Training Cloud Environment and login as "ACLBizet"
2. Open the Batch_Report -> Tools > Report Version Manager… -> Review -> enter Comments -> Accept (or Reject to reject Review process and return to In Edit) -> enter user Password -> OK -> Close. **Note:** the report is now released for users with privileges



Commenting report (optional for approving report)

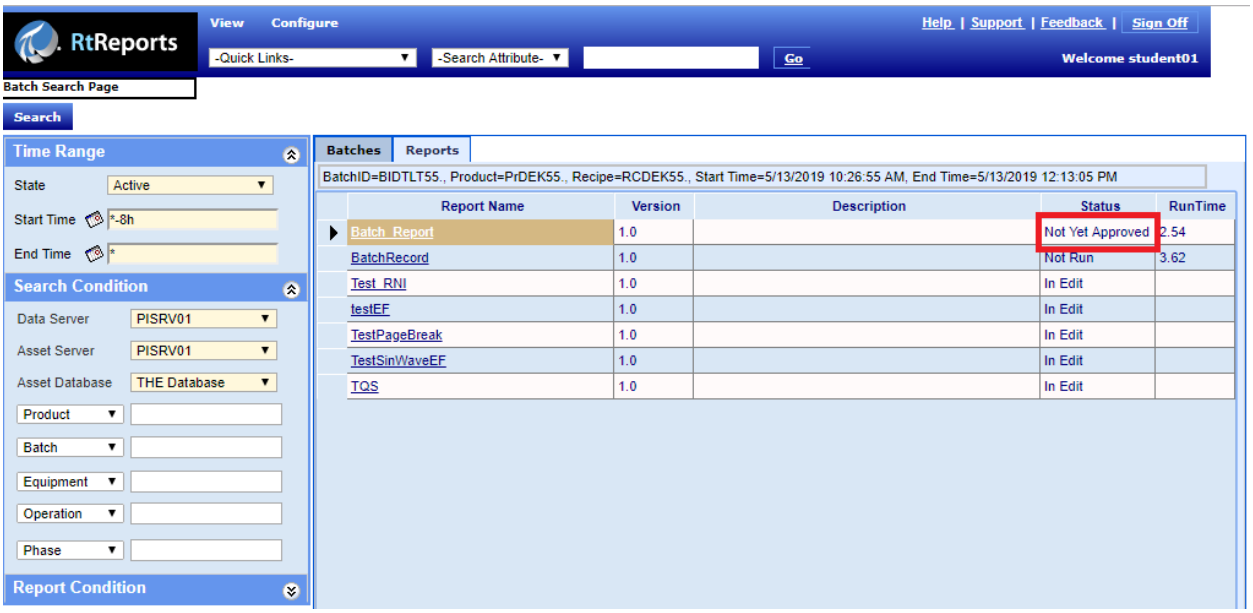
1. Launch RtReport Generator on the PIRPT01 Training Cloud Environment and login as "LBernstein"
2. Fill in Time Range and Search Condition -> Search -> find and select a completed (with Start and End Times) BIDxxx EF (**record the Batch ID, Start Time, and End Time**) -> Reports tab -> find and select the Batch_Report batch report template hyperlink (notice that the Status: **Not Run**)



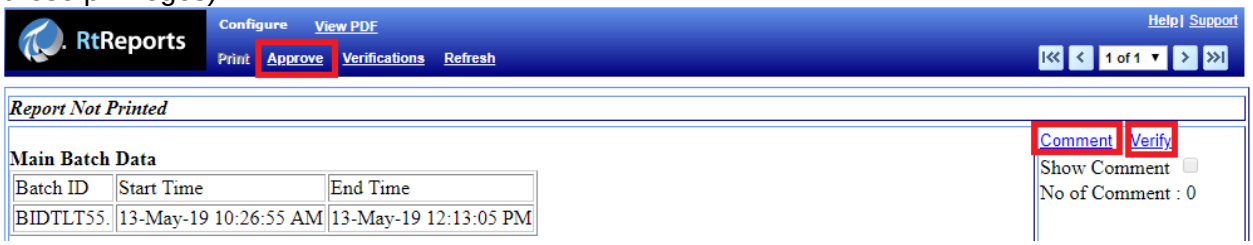
3. In the generated report -> click *Comment* hyperlink -> enter comments…mment" Hyperlinks > enter comments… Notice that the *Verify* hyperlinks and *Approve* button are disabled (because LBerstein does not have those privileges)
4. Sign Off RtReports Generator

Verify report (optional for approving report)

1. Launch RtRports Generator on the PIRPT01 Training Cloud Environment and login as "WAMozart"
2. Find and select the EF and re-generate the Batch_Report (same as above). **Notice**, the report status "Not Yet Approved"



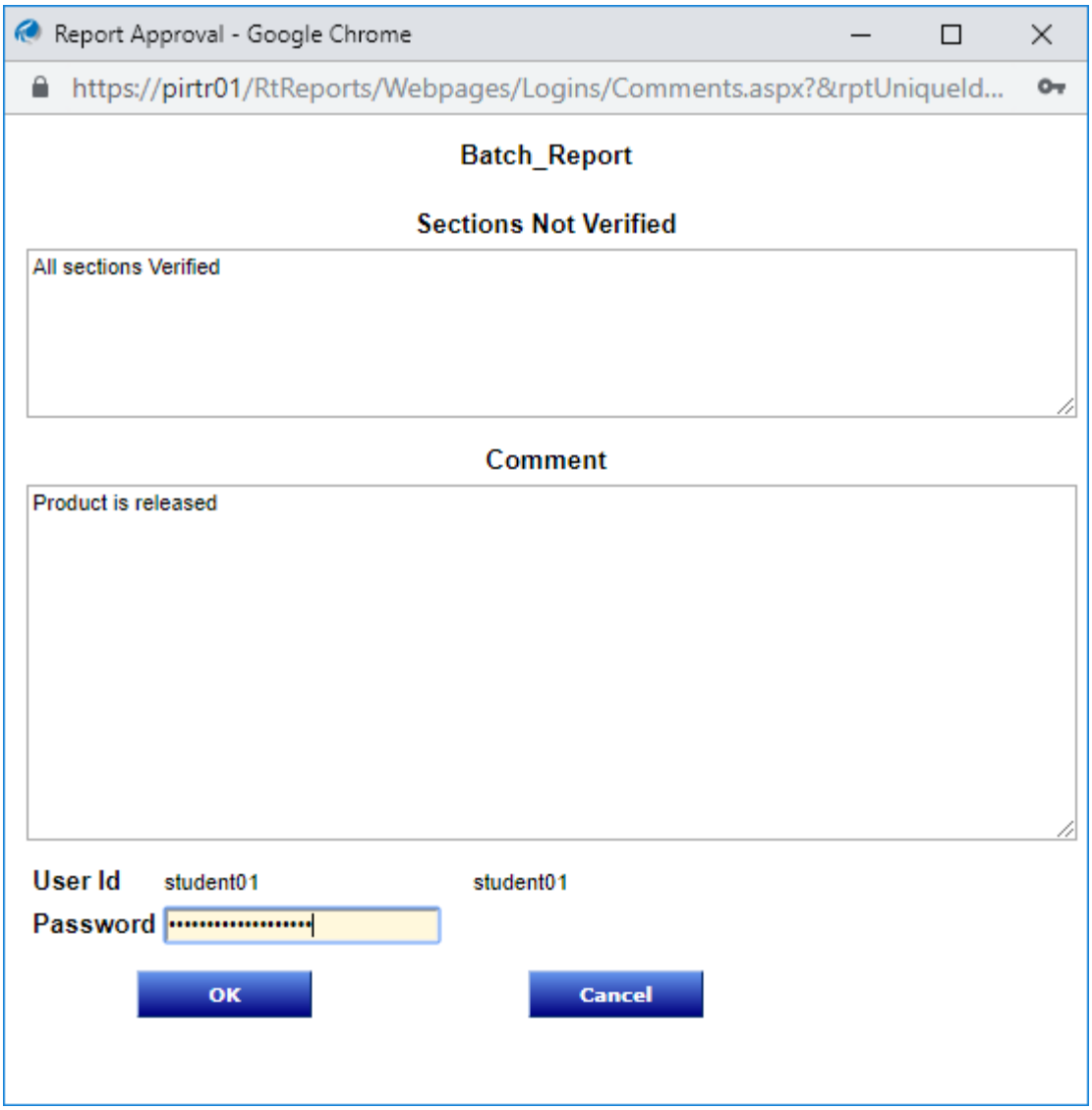
3. In the generated report page, click the ***Verify*** hyperlink -> enter verifications… Notice the Comment hyperlinks and Approve button are disabled (because WAMozart does not have those privileges)



4. Sign Off RtReports Generator

Approve report

1. Launch RtReport Generator on the PIRPT01 Training Cloud Environment and login as "LBeethoven"
2. Find and select the EF and re-generate the Batch_Report (same as above). **Notice**, the report status "Not Yet Approved"
3. In the generated report, click on the *Approve* -> enter comment… -> OK. **Notice,** the official print "Print" button is enabled now



4. Notice the ***Print*** hyperlink is enabled; to obtain an Official report hard copy, click on the ***Print*** hyperlink. **Official report(s)** is in C:\AdobePDF_Official directory
5. To obtain an **unofficial report** hard copy, simply click on the View PDF hyperlink. Un-official report(s) is in C:\AdobePDFLocation directory. Note: users who has the Un-official Print privileges may obtain an un-official report hard copy whether the report is approved or not approved

Lesson 11: Miscellaneous Topics

PI Batch Interfaces

Batch systems using Batch Execution System (BES) records batch process events in a file of .evt extension and/or in SQL database during execution of batch recipe.

PI Batch Interface is the next generation of “evt File Interface”; it is a “company” specific batch system interface (a given PI Batch Interface can only talk to a company’s batch system).

Example:

- PI Batch Generator
- PI Event Frame Generator
- ABB800xA for ABB Batch
- EMDVB for Emerson DeltaV Batch
- EMDVBCS for Emerson DeltaV Syncade
- FTBInt for Rockwell FactoryTalk Batch
- GEIB for GE iBatch
- HWTPB for Honeywell Total Plant Batch
- WPASXBatch for Werum PAS-X Batch
- WWInBatch for WonderWare InBatch

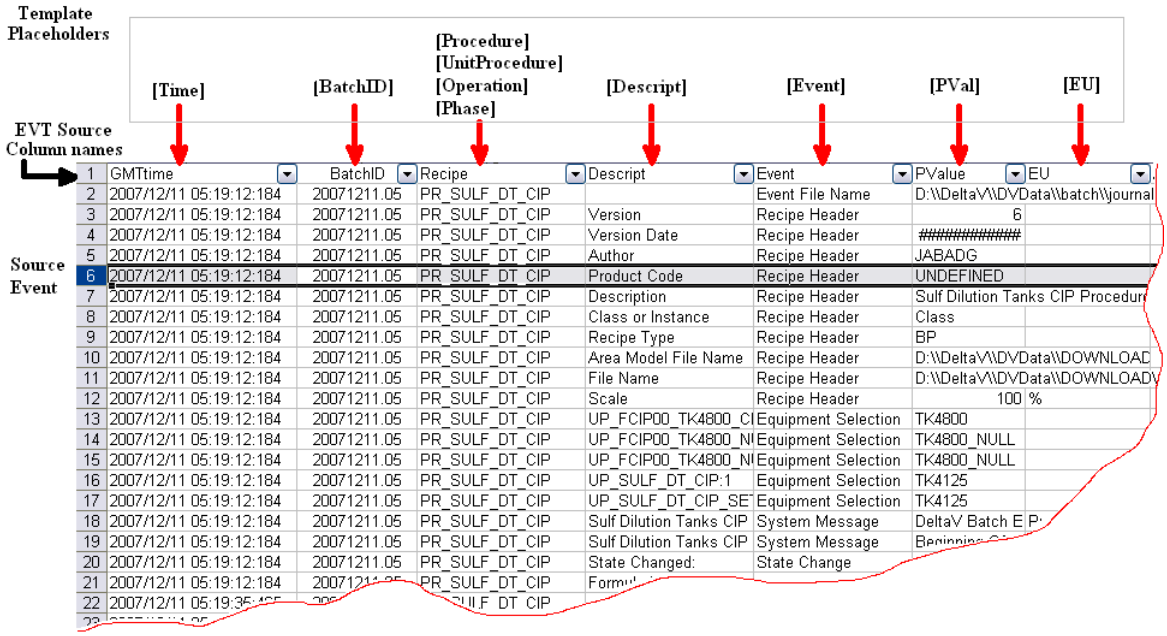
PI Batch Interface is responsible for collecting batch process events and automatically populate PI Batches or AF EventFrame.

Depending on users’ Batch Execution System, PI Batch Interface may be configured to collect batch process events from .evt files or SQL database. Also, user have the options of populating PI Batch that creates PI Modules, PI Tags, PI Aliases, and PI Batches automatically or AF Event Frame that creates AF Elements, PI Tags, AF Attributes, and AF Event Frame automatically. **Note:** PI Batch Interface no longer renames .evt file when encounters “End of Batch”.

Template Placeholders

PI Batch Interface uses [TAG TEMPLATE] and [PROPERTY TEMPLATE] to specify what is stored in PI Batch Database, PI Module Database (create PI Modules and PI Aliases automatically), AF Event Frame, AF Element (create AF Elements and AF Attributes automatically), and PI Points. Templates define the custom name and/or value structure applied to a particular PI object. Users must define TEMPLATE

using free texts and placeholders (names of columns in an .evt file or SQL database).



	GMTtime	BatchID	Recipe	Descript	Event	PValue	EU
2	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP		Event File Name	D:\\DeltaV\\DVData\\batch\\journal	
3	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Version	Recipe Header	6	
4	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Version Date	Recipe Header	##########	
5	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Author	Recipe Header	JABADG	
6	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Product Code	Recipe Header	UNDEFINED	
7	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Description	Recipe Header	Sulf Dilution Tanks CIP Procedur	
8	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Class or Instance	Recipe Header	Class	
9	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Recipe Type	Recipe Header	BP	
10	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Area Model File Name	Recipe Header	D:\\DeltaV\\DVData\\DOWNLOAD	
11	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	File Name	Recipe Header	D:\\DeltaV\\DVData\\DOWNLOADV	
12	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Scale	Recipe Header	100	%
13	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	UP_FCIP00_TK4800_CI	Equipment Selection	TK4800	
14	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	UP_FCIP00_TK4800_NI	Equipment Selection	TK4800_NULL	
15	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	UP_FCIP00_TK4800_NI	Equipment Selection	TK4800_NULL	
16	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	UP_SULF_DT_CIP:1	Equipment Selection	TK4125	
17	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	UP_SULF_DT_CIP_SE	Equipment Selection	TK4125	
18	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Sulf Dilution Tanks CIP	System Message	DeltaV Batch E	P
19	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Sulf Dilution Tanks CIP	System Message	Beginning Ci	
20	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	State Changed:	State Change		
21	2007/12/11 05:19:12:184	20071211.05	PR_SULF_DT_CIP	Formul			
22	2007/12/11 05:19:35:485	2007	ULF_DT_CIP				

Example:

[TAG TEMPLATE]

TAG[8000101].NAME=Building-10:Event.Recipe_Value_Change

TAG[8000101].VALUE=[BATCHID]|[UNIQUEID]|[PROCEDURE]\[UNITPROCEDURE]\[OPERATION]\[PHASE]|[DESCRIPT]|[EVENT]|[PVAL]|[EU]|[AREA]|[PROCESSCELL]|[UNIT]|[PHASEMODULE]|[USERID]|

TAG[8000101].TYPE=string

TAG[8000101].TRANSLATE=False

TAG[8000101].TRIGGER=[EVENT,value="Recipe Value Change"]

[PROPERTY TEMPLATE]

PROPERTY[10].VALUE=[TIME]|[DESCRIPT]|[EVENT, value="Formula Header"]|[PVAL]|[EU]|[AREA]|[PROCESSCELL]|[UNIT]|[PHASE]|[PHASEMODULE]|[USERID]|[UNIQUEID]

PROPERTY[10].TYPE=String

PROPERTY[24].VALUE=[TIME]|[DESCRIPT]|[EVENT, value="Recipe Header"]|[PVAL]|[EU]|[AREA]|[PROCESSCELL]|[UNIT]|[PHASE]|[PHASEMODULE]|[USERID]|[UNIQUEID]

PROPERTY[24].TYPE=String

Refer to desired PI Batch Guide for detail.

PI Baseline Service

Introduction

PI Baseline Service (formerly RtBaseline Services) is the application layer that RtReports uses to access its data sources. It is a data foundation for RtReports.

Any data source or calculation used in the reports must be configured in PI Baseline Services first. Only configured data sources are exposed.

All configuration is performed through a web browser, the link to the page is:

```
http://<<server name>>/RtReports/Admin
```

where <<server name>> is the host name or IP Address of user's RtReports Server.

Note: The ability to edit Data Sources or DataSets is controlled by membership in the local server group RtBaselineAdmins.

Data Sources

The type of data source: Relational Data Source

Relational Databases are often used to get non-time series data. This might include:

- Recipe/Ingredients
- Operator's Comment
- Calibration Data
- Lab Data
- ERP Data
- Maintenance Data

Note: Each PI Server used as a data source must have a trust configured.

The trust can either be to the RtReports server machine, the application used (w3wp.exe), or to each specific user accessing data. Tag level security is preserved through PI Baseline Services.

Relational Databases will require a connection string and may have their own security requirements. Consult the documentation for that application for details.

Server side Configuration Files

System defaults settings are stored in multiple XML Configuration Files.
Customization of RtReports is done by modifying these XML Configuration Files.
Complete documentation can be found in the RtReports Administrator Guide.

Caution: Make a backup of the XML Configuration File before editing them.
Consult a technical support agent if you are in doubt.

Configuration Files

The table below lists the locations of the configuration files that control some of the aspects of the RtReports application.

System Defaults Grouping	Configuration File
RtReports Server Defaults	\PIPC\RtReports\Web.Config
RtReports Remoting Cache Manager Defaults	\PIPC\RtReports\Web.Config
RtReports Remoting RtReportsRunner Defaults	\PIPC\RtReports\Web.Config
Report Remoting Generator Defaults	\PIPC\RtReports\Web.Config
RtReports Remoting Cache Manager Defaults	\PIPC\RtReports\Web.Config
RtReports Editor Defaults	\PIPC\RtReports\Common\OSIsoft. RtReports.Editor.exe.config

Server Defaults

Parameters	Information
DefaultStartTime	Controls the default query time used in the Generator
AllowCommentandVerify	True or False AllowCommentandVerify AllowCommentAfterApprove AllSectionVerifiedToApprove AllowMultipleVerification
eMailErrorTo	Web site error are e-mailed to this address
InitialBatchSearch	True or False
InitialPage	Batch (default) or Report
ReportTimeZone	Server (default) or Client
SnapshotDelayTime	Amount of time after the end time of a report context before that context is deemed complete
ReportRequestInterval	Refresh interval for the report page in seconds
ImageFileSettings	Allow to add an image on the logon page

Editor Defaults

Parameters	Information
Server	Server on which the templates are stored
TemplatePath	The Module Database path to the base RtReports module
PagePath	The base URL to RtReports Web application
RemoteServerName	RtReports Server name
RemoteServer	RtReports Server name and path for the Web wervices definitions

Report Execution Manager and Report Cache

The Report Execution Manager is a new RtReports server module created to manage the execution of reports. This module will monitor 4 queues: a report cache queue, a report running queue, a report pending queue, and a report error queue.

The Report caching mechanism has also be expanded. A new RtReports server module was created to provide a configurable file persisted report cache.

All configuration is performed through a web browser, the link to the page is:

https://<<server name>>/RtReports/WebPages/Cachemanagement.aspx

where <<server name>> is the host name or IP Address of your RtReports Server.

Parameters	**Information**
StorageSize	Number of reports cached before they are purged (100)
Directory	Location of the cached reports
ReloadCache	Controls whether the existing cached reports are reloaded from the files when the RtReportRunner and CacheManager are restarted (TRUE)

The Cache is defined in:

%PIHOME%\RtReports\Web.config

Cache folder defined by the Directory key

```
<add key="Directory" value="%PIHOME%\RtReports\Cache">
```

Persistence of the cache

```
<add key="ReloadCache" value="true">
```

Set to false, it would force the RtReports to ignore the cached reports upon restarts.

Batch Reference Information

PI Batch Database Architecture

Introduction

The PIBatch/EF Database architecture consists of the following levels:

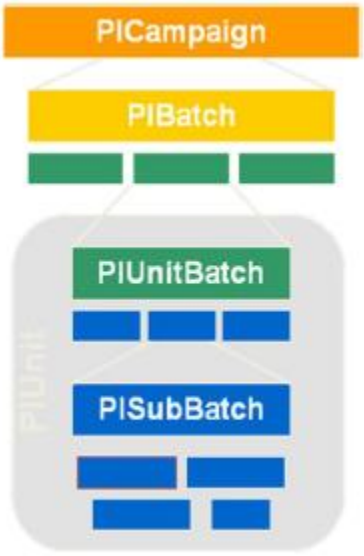
- PICampaign
- PIBatch
- PIUnitBatch
- PISubBatch
- PISubBatche

The PICampaign level has not been implemented.

What is the PI Batch Database?

The PI Batch Database is a subset of the PI Module Database and was introduced in PI Server 3.3. It enables real time event data archiving and presentation of Campaigns, Batches, Unit Batches and Sub-Batches.

It makes use of the Module Database **Aliases** for common name references for tags and **Properties** for configuration data. It allows complete read/write access. It is logically structured as in the illustration below:



A PIBatch is made up of one or more PIUnitBatches. A PIUnitBatch can have one or many layers of PISubBatches.

PIBatch

The PIBatch/Procedure object is the highest-level time series object used to record and track batch production. Batch in this case, is the material made--not the physical processing required in making the material. PIBatch objects do not directly relate to production equipment; PIUnitBatch objects handle this.

PIBatch Properties

Here's a description of each property of the PIBatch.

- StartTime/EndTime
- Batch ID
- Product
- Recipe

Each PIBatch or Procedure can contain one or more PI UnitBatches or Unit-Procedures

PIUnitBatch

PIUnitBatch objects represent a period of processing or activity in a specific piece of equipment represented by a special module in the PI Module Database.

***Only one PIUnitBatch may be active in a PIUnit over any time period*.** The StartTime of a batch may equal the EndTime of the previous batch; no other overlap is allowed.

PIUnitBatch or Unit-Procedure Properties

- StartTime/EndTime
- UnitBatch ID
- Product
- Recipe / Procedure
- Associated PIBatch or Procedure

Each PI UnitBatch can contain one or more PI SubBatches….

PISubBatch

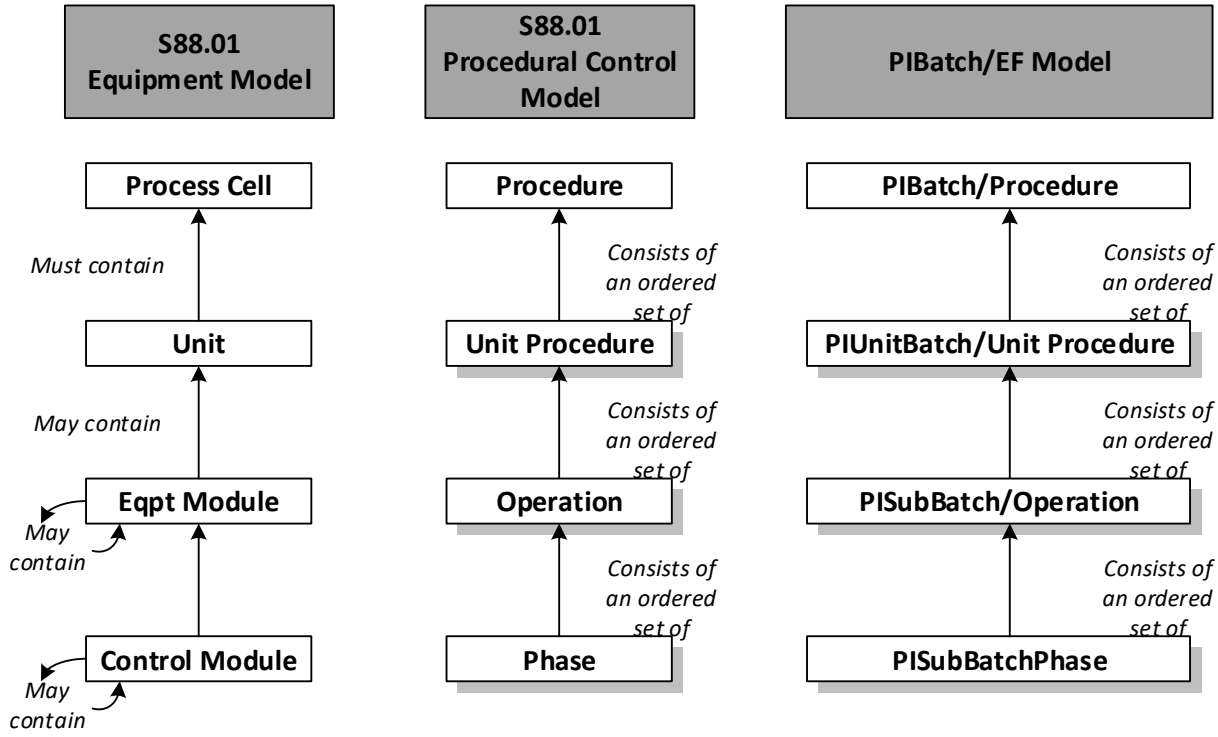
A PISubBatch represents an identifiable portion of a PIUnitBatch. Examples of sub-batches are Operation, Phase, and Step.

PISubBatch objects allow special identification to portions of a PIUnitBatch. This collection is hierarchical, thus allowing collection of PISubBatch objects within other PISubBatch objects. For example, it is possible to have 2 simultaneous (parallel) phases and explicitly assign steps to the appropriate phase.

S88 examples of sub-batch classification or types are Operation, Phase, and Step. This is common terminology that may not be applicable everywhere. For flexibility, the PISubBatch object has a PI Heading property. This allows creation of PI HeadingSets, which represent sub-batch classification names.

How PIBatch/EF Maps to the S88 Standard

PIBatch/EF does not directly adhere to a specific mapping to the S88 standard. However, a common mapping is show below:



However, the user is flexible to use any structure that is desired for the business.

Examples of Batch

PI Batch can be used to frame any repeatable time segment for which the start and end time can be used for further process analysis:

- Uptime/downtime tracking (equipment, production…)
- Transition time between production runs
- Grade transition tracking
- Finished product or grade tracking
- Raw material input tracking
- Environment regulation tracking (out of compliance)
- Performance vs. production team tracking
- Track production according to financial calendar

Take a few moments and list some of your own applications of a Batch or Event Framing System:

Exercise: BatchView

Problem Description

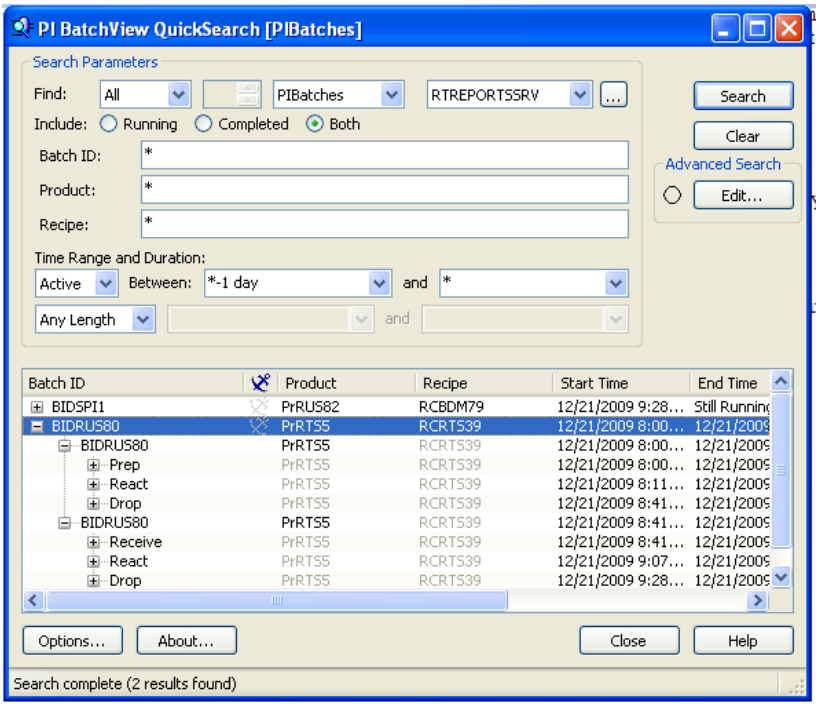
You need to see some batch data to determine if you have data for building a report. You may do any one of the following:

Approach

Launch the PI BatchView Quick Search (installed with RtReports Server and RtReports Editor).

Search PIBatches of Batch ID stem “BID” for the last 1 day.

Examine the results of the query.



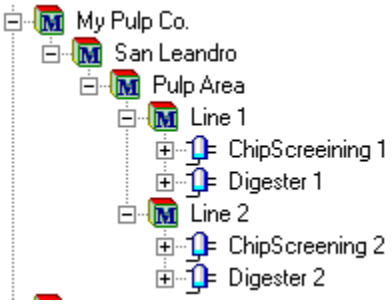
PI Module Database

The PI Module Database can benefit your users by providing an easier way to find and view data. The two features that allow that are Aliases and Properties.

Modules

Modules can be built in any hierarchical structure. They can be built based on organization, geography, product line, or any criteria. The best hierarchy is one that the users will find most helpful.

Caution: Do not build Modules in or modify the items in the %OSI module.



Sample structure based on plant layout

Each module can contain an unlimited number of Aliases and Properties.

Note: Any user or application that is creating or editing batch data in PI will require write access to the PI Module Database. This is normally accomplished for applications using a trust although this is not required.

Heading Sets

A PIHeading Set is optional and provides a collective name for each level of PISubBatches defined in the SubBatch Hierarchy. For example, according to the S88 standards, the first level of PISubBatches is called 'Operations' and the second level is called 'Phases'. To provide flexibility in naming these levels, PIHeading Sets are used. A PIHeading Set consists of PIHeadings and a level corresponding to the PIHeading. There can be only one PIHeading at each level. Therefore, if a PIHeading Set consisting of PIHeadings called 'Operations', 'Phases' and 'Steps' at levels 1, 2, and 3 respectively is specified for a particular PIUnit, it means that all the PISubBatches defined at the first level are Operations, all the PISubBatches defined at the second level (does not matter which PISubBatch is their parent PISubBatch)

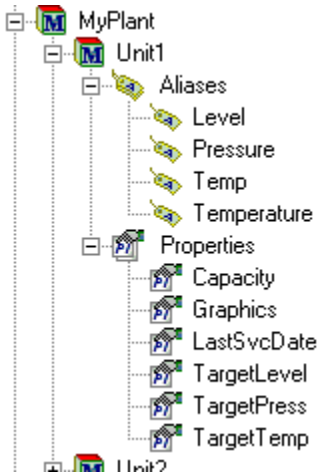
are Phases and all the PISubBatches defined at third level are Steps. If there is no PIHeading Set specified or if there is no corresponding PIHeading for any level, this attribute of the PIHeading Set is left empty.

Aliases

Aliases are common names for tags. Especially useful is when you have many units that have the same functions (aliases). For example, if you had many turbines and they all had tags that measure temperature, oil pressure, rpm, etc.

Properties

Properties are aspects that do not change often. Examples of this might be height, radius, high limit, installation date / technician.



Example of aliases and properties

PI Module Database Structures Used in PI Batch

The structure of the PI ModuleDB allows the information to be stored hierarchically in relation to a physical model, a business model or an application model.

You must build a Module Database structure to support your Batch structures. Whether you elect to build a structure based on the S88 Physical Model of your plant or site or you use a Procedural Model is irrelevant to the PI System.

You should build your structures based on how you think your users will most easily want to organize and find the Batch data at a later time.

PI Module Database Tools

There are two primary tools used to create and maintain entries in the PI Module Database. The most common is the PI Module Database plug-in in the SMT kit. The other is the PI Module Database Builder add-in to Microsoft Excel.

Exercise: Configure *PI Modules* for Pulp and Paper Line 1

Objectives

Discover your PI Module Database Structure.

Problem Description

Configure “Pulp and Paper” production “Line 1” which consists of Chip1 unit and Digest1 unit.

Add Aliases to each PI Units.

Approach

Open PI System Management Tools (PI SMT) and browse to Module Database

Add a *root PI Module* “Pulp and Paper”

Add a *child PI Module* “Line 1” to “Pulp and Paper”

Add a *child PI Unit* “Chip1” to PI Module “Line 1”

Add a *child PI Unit* “Digest1” to PI Module “Line 1”

Add an Alias “Pres” (PI Tag: PP:Digest1.Pres) to PI Unit “Digest1”

Add an Alias “Temp” (PI Tag: PP:Digest1.Temp) to PI Unit “Digest1”

Add an Alias “Weight” (PI Tag: PP:Chip1.Weight) to PI Unit “Chip1”

Exercise: Configure PI Batch for Pulp and Paper Line 1

Objectives

Learn how to configure PI Batch using PI Batch Generator (PIBaGen).

Problem Description

Configure PI Batches for Pulp and Paper Process, production Line 1 which consists of a Chip unit and a Digest unit.

Approach

Open PI SMT and browse to Batch Generator
Register Chip1 and Digest1
Configure PIUnitBatches for PI Unit Chip1
- Active Point: PP:Chip1.AP
- ActivePoint Behavior: Pulse
- Unit Batch ID Point: PP:Chip1.UBID
- Product Name Point: PP:Chip1.Prod
- Procedure Name Point: PP:Chip1.Proc
- Evaluation Delay: 5 seconds

Configure PIBatches for PI Unit Chip1
- PIBatch Index Point: PP:Pulp1.BID
- PIBatch Product Point: PP:Pulp1.Prod
- PIBatch Recipe Name Point: PP:Pulp1.Rec

Configure PIUnitBatches for PI Unit Digest 1
- Active Point: PP:Digest1.AP
- ActivePoint Behavior: Pulse
- Unit Batch ID Point: PP:Digest1.UBID
- Product Name Point: PP:Digest1.Prod
- Procedure Name Point: PP:Digest1.Proc
- Evaluation Delay: 5 seconds

Configure PISubBatches “OP” for PI Unit Digest 1
- Active Point: PP:Digest1.OP
- ActivePoint Behavior: Step
- SubBatch Name: Use ActivePoint value

Configure PISubBatches “PH” for PI Unit Digest 1
- Active Point: PP:Digest1.PH
- ActivePoint Behavior: Step
- SubBatch Name: Use ActivePoint value

Exercise: Configure PI Batches for Pulp and Paper Line 2

Problem Description

Discover your PI Module Database Structure.

Configure “Pulp and Paper” production “Line 2” which consists of Chip2 unit and Digest2 unit.

Add Aliases to PI Units.

Approach

Refer to steps in Directed Exercise – Configure PI Module for Pulp and Paper Line 1 and Directed Exercise – Configure PI Batch for Pulp and Paper Line 1

PI Units and Aliases Configuration

Unit Name	Alias Name	PI Point
Chip2	Weight	PP:Chip2.Weight
Digest2	Pres	PP:Digest2.Pres
	Temp	PP:Digest2.Temp

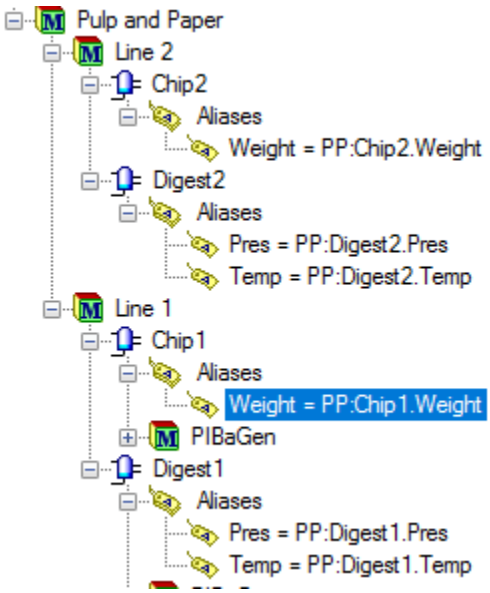
PI Batch Configuration

Unit	Batch Level	Name	Active Point	Batch ID Point Product Point Procedure Point
Line 2	Procedure			PP:Pulp2.BID PP:Pulp2.Prod PP:Pulp2.Rec
Chip2	Unit Procedure		PP:Chip2.AP	PP:Chip2.UBID PP:Chip2.Prod PP:Chip2.Proc
Digest2	Unit Procedure		PP:Digest2.AP	PP:Digest2.UBID PP:Digest2.Prod PP:Diges21.Proc
	Operation	OP	PP:Digest2.OP (Step)	
	Phase	PH	PP:Digest2.PH (Step)	

Solution: Configure PI Batch for Pulp and Paper Line 2

Create new PI Modules/PI Units

1. From PIRPT01 Training Cloud Environment, Open PI SMT -> Operation -> Module Database
2. Find *PI Module* “Pulp and Paper” -> right click -> New -> Module -> Name “Line 2
3. Right click *PI Module* “Line 2” -> New -> PIUnit -> Name “Chip2”. Repeat for PI Unit “Digest2”
4. Right click *PI Unit* “Chip2” -> New -> Alias -> Name “Weight”, PI Point: PP:Chip2.Weight
5. Right click *PI Unit* “Digest2” -> New -> Alias -> Name “Pres”, PI Point: PP:Digest2.Pres
   Repeat for *alias* “Temp”, PI Point: PP:Digest.Temp



Configure PI Batches

1. Open PI SMT -> Batch -> Batch Generator -> *MDB View* tab
2. Find and expand *PI Module* “Line 2”
3. Right click *PI Unit* “Chip2” -> Register. Repeat for *PI Unit* “Disgest2”
4. Select tab *Registered Units Only*
5. Find and select PI Unit “Chip2”
   - Configure PIUnitBatches
     - Active Point: PP:Chip2.AP
     - ActivePoint Behavior: Pulse
     - Unit Batch ID Point: PP:Chip2.UBID
     - Product Name Point: PP:Chip2.Prod
     - Procedure Name Point: PP:Chip2.Proc
   - Configure PIBatches
     - PIBatch Index Point: PP:Pulp2.BID
     - PIBatch Product Name Point: PP:Pulp2.Prod
     - PIBatch Recipe Name Point: PP:Pulp2.Rec

6. Find and select PI Unit “Digest2”
   - Configure PIUnitBatches
     - Active Point: PP:Digest2.AP
     - ActivePoint Behavior: Pulse
     - Unit Batch ID Point: PP:Digest2.UBID
     - Product Name Point: PP:Chip2.Prod
     - Procedure Name Point: PP:Chip2.Proc
   - Configure PIBatches -> *same as Chip2’s*
   - Configure PISubBatches “OP”
     - Active Point: PP:Digest2.OP
     - ActivePoint Behavior: Step
     - SubBatch Name: Use ActivePoint value
   - Configure child PISubBatches “PH” for SubBatch “OP”
     - Active Point: PP:Digest2.PH
     - ActivePoint Behavior: Step
     - SubBatch Name: Use ActivePoint value
7. SAVE!!!

Exercise: Configure PI Batch for Cake Making

Objectives

Configure PI Batches for Cake Making Process which consists of a Cake unit and a Crust unit.

Approach

Refer to steps in Directed Exercise – Configure PI Module for Pulp and Paper Line 1 and Directed Exercise – Configure PI Batch for Pulp and Paper Line 1

PI Units and Aliases Configuration

Element Name	Attribute Name	PI Point
Cake	Cheese Amount	CCF:Cake.Charge.Cheese.Qty
	Eggs Amount	CCF:Cake.Charge.Eggs.Qty
	Flour Amount	CCF:Cake.Charge.Flour.Qty
	Liquid Amount	CCF:Cake.Charge.Liquid.Qty
	Sugar Amount	CCF:Cake.Charge.Sugar.Qty
	Vanilla Amount	CCF:Cake.Charge.Vanilla.Qty
	Yolks Amount	CCF:Cake.Charge.Yolks.Qty
Crust	Butter Amount	CCF:Crust.Charge.Butter.Qty
	Cinnamon Amount	CCF:Crust.Charge.Cinnamon.Qty
	Crackers Amount	CCF:Crust.Charge.Crackers.Qty
	Nuts Amount	CCF:Crust.Charge.Nuts.Qty

PI Batch Configuration

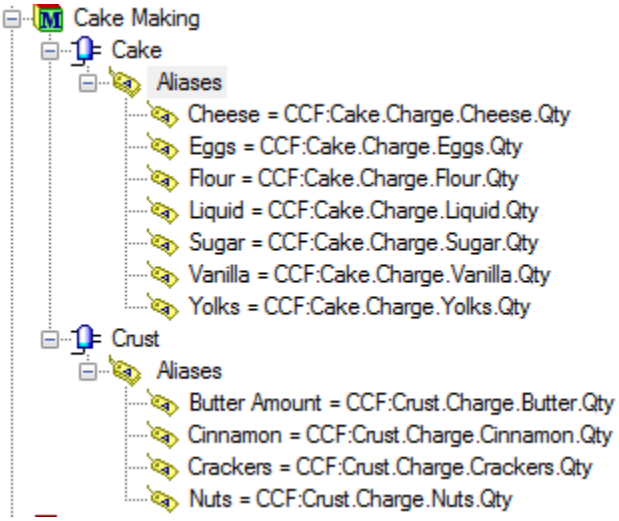
Unit	Batch Level	Name	Active Point	Batch ID Point Product Point Procedure Point
	Procedure			CCF:BID
Cake	Unit Procedure		CCF:Cake.AP	CCF:Cake.UBID CCF:Cake.Prod CCF:Cake.Proc
	Operation	Charge	CCF:Cake.Charge	
	Phase	Cheese	CCF:Cake.Charge.Cheese	
		Eggs	CCF:Cake.Charge.Eggs	
		Flour	CCF:Cake.Charge.Flour	
		Liquid	CCF:Cake.Charge.Liquid	
		Sugar	CCF:Cake.Charge.Sugar	
		Vanilla	CCF:Cake.Charge.Vanilla	
		Yolks	CCF:Cake.Charge.Yolks	
	Operation	Mix	CCF:Cake.Mix.Operation	
	Phase	Mix (step)	CCF:Cake.Mix.Phase	
	Operation	Discharge	CCF:Cake.Discharge	
Crust	Unit Procedure		CCF:Crust.AP	CCF:Crust.UBID CCF:Crut.Prod CCF:Crust.Proc
	Operation	Charge	CCF:Crust.Charge	

	Phase	Butter	CCF:Crust.Charge.Butter	
		Cinnamon	CCF:Crust.Charge.Cinnamon	
		Crackers	CCF:Crust.Charge.Crackers	
		Nuts	CCF:Crust.Charge.Nuts	
	Operation	Mix	CCF:Crust.Mix	
	Operation	Discharge	CCF:Crust.Discharge	

Solution: Configure PI Batches for Cake Making

Consult Exercise – Configure PI Batches for Pulp and Paper Line 2 steps.

Create new PI Modules/PI Units



Configure new PIBatches for Cake Making

1. PI SMT -> Batch -> Batch Generator -> MDB View tab
2. Register PI Units Cake and Crust
3. Registered Units Only tab
4. Select *PI Unit* Cake
    - Configure PIUnitBatches
    - Configure PIBatches
    - Configure PISubBatches and PISubBatches for each PISubBatches
5. Select *PI Unit* Crust
    - Configure PIUnitBatches
    - Configure PIBatches
    - Configure PISubBatches and PISubBatches for each PISubBatches

SAVE!!!

PI AF, Asset Analysis and Event Frames (EF)

What does Asset Framework (AF) do for me?

The *PI Data Archive* is focuses on a points database and is extremely good at storing vast amounts of data collected by interfaces. It allows easy and performant retrieval of time-series data. The PI Data Archive architecture is scalable, maintainable and highly available.

The *Asset Framework (AF)* supplements the architecture by providing a Meta-data structure for all data in the organization ("Data Directory"). Asset Framework (AF) has a rich set of features and functions to organize and enhance the data in the PI Data Archive. Because it offers user-friendly access to the data it is the preferred way for users to interact with their PI system data.

What are the benefits of Asset Framework?

Easy way to navigate throughout the system

The hierarchical asset structure gives a convenient way for navigation that can also be used by people, who are not familiar with the technical details of how the data is retrieved from the data source.

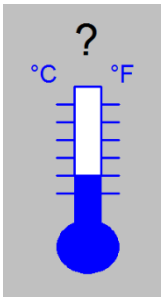
Unify data from disparate source systems

PI Point attributes get their data from PI Data Archives. AF is not limited to one PI Data Archive. Attributes can refer to multiple PI Data Archive Servers (either standalone or collective)

The origin of data can also be a relational database. While some attributes of an element representing a reactor are time-series data coming from PI tags (such as the temperature), data for some other attributes can be from an external database (such as the physical characteristics or inventory data).

All attributes are listed side by side giving the user a comprehensive insight into all the relevant data of the asset.

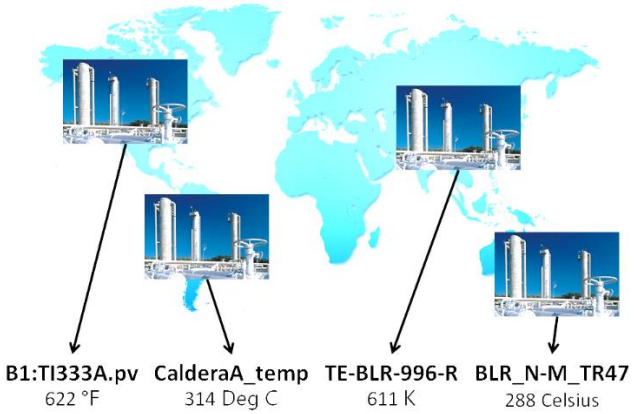
Supports different Units of Measure



PI Asset Framework (PI AF) attributes are associated with specific Units of Measure. AF is preloaded with numerous standard unit-of-measure classes and conversion factors based on the International System of Units (SI). It also supports user defined classes and Units of Measure. For example, the source unit for a temperature sensor attribute in a reactor in the USA can be associated with degrees Fahrenheit, the corresponding attribute for a reactor in Italy can be associated with degrees Celsius. When working with the data (doing calculations or displaying the data) this is properly taken into account.

Standardization with Templates

At many sites there are no rigorous naming standards for the points. There may be missing descriptions and/or engineering units. The PI System is often used to integrate information from different sources and these sources may not have been configured consistently:



Element templates in AF provide the basis for standardization. When applied for elements that represent the same type equipment, all elements have the same set of attributes with a consistent, user friendly naming. The attributes have same unit of measurement, same data type, the same description, etc. This allows a harmonized, consistent representation of your system.

Another benefit of using templates is the quick creation of many elements of the same type.

Powerful Calculation Options (Asset Analytics)

Users can configure, schedule and run calculations written using PI Performance Equation (PE) syntax acting on their PI Asset Framework (PI AF) attributes. PE expressions, Rollup calculations and generation of PI Event Frames based on trigger conditions are all supported analysis types. In addition, analysis templates enable users to manage their analyses in a standardized and consistent manner. Typical applications are Key Performance Indicator (KPI) calculations and condition-based maintenance (CBM).

Key Features of the AF Server

Scalability

- Up to **10 million assets** with references to up to a hundred Data Archive points across multiple PI Systems.
- Hosted in **32-bit or 64-bit** Microsoft operating systems.
- Compatible with **Microsoft SQL Server** Express and up to Microsoft SQL Server farms with mirroring.

Usability

- **Flexible hierarchy** and powerful search and indexing support for finding data.
- Integrated easy-to-use **calculations and lookup tools**.
- Powerful **templates** support discipline and maintenance for relationships and calculations.
- **Integrated to our visualization tools** (PI Coresight, PI Datalink, PI ProcessBook and PI WebParts).
- Microsoft integrated and validated **Windows security**.

Extensibility

- Supported by **Notifications** alerting and event filtering application.
- Supported by **PI OLEDB Enterprise** and **PI Web Services** query capabilities with flexible SQL and XML standards.
- Published **AFSDK** (AF) and ANSDK (Notifications) with training and support on **OSIsoft vCampus**.
- Customers and partners can create **custom** calculations (data references) and filtered data recipients (delivery channels) with Microsoft Visual Studio (C# or VB).

Reliability

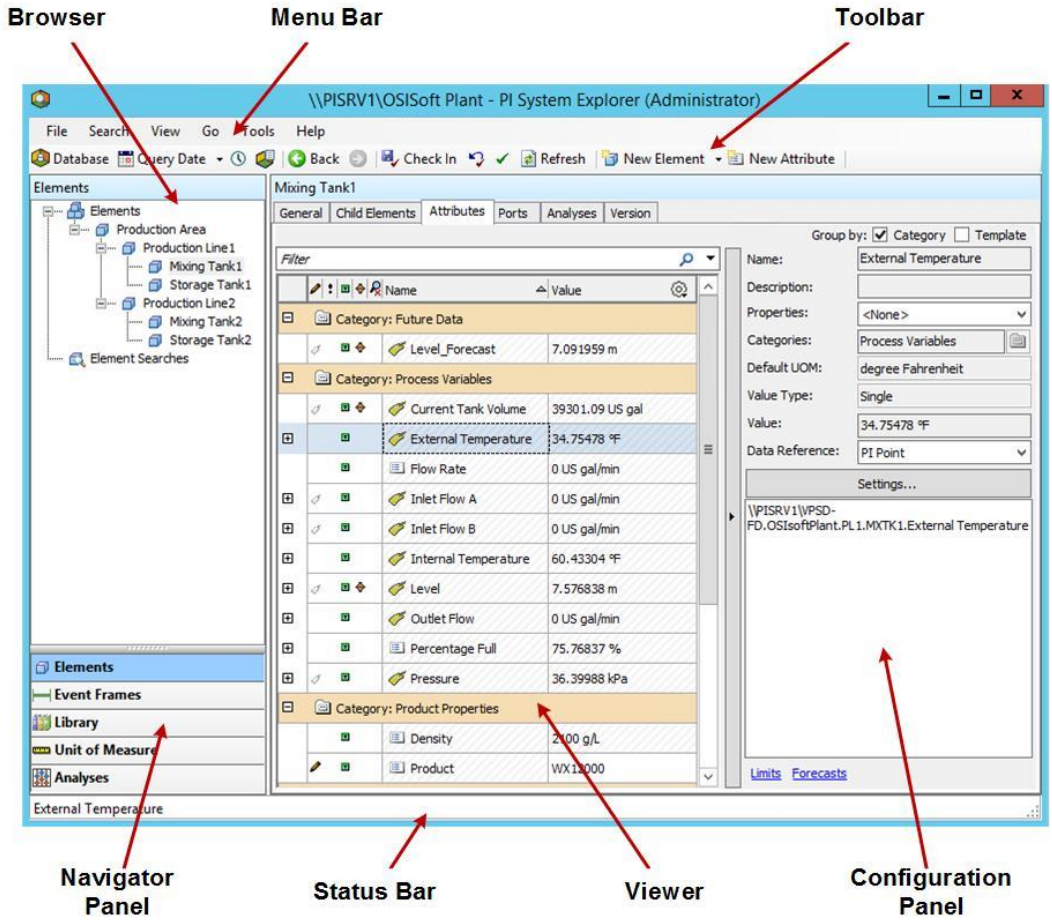
- AF configured for **High Availability** (HA) assures the AF directory and information is always available.
- Compatible with our proven **PI System HA** technology.
- Worldwide quality **technical support**.
- Disciplined store for your enterprise's process knowledge offering versioning, backup, detailed integrated security and record locking collaboration.

PI System Explorer

PI System Explorer, or sometimes referred to as PSE or AF Client, is the AF user interface and allows users to find information about their equipment and processes. It also has a rich range of features, making it the configuration and management tool for AF, PI Notifications and Event Frames.

PSE is installed as part of the PI ProcessBook and PI DataLink installations. The client component of the PI Notifications install kit will add extra sections to the Navigator Panel of PSE (MyPI, Notifications and Contacts) in order to configure notifications.

The major components of the PSE are shown in the following:



Menu Bar / Toolbar

Use these bars for tasks such as opening/creating a database, searching for elements or contacts, applying and checking in changes, setting view options, and more. Menus and the Toolbar are context sensitive and will present different options depending on what section is selected from the Navigator Panel.

Navigator Panel

PI System objects are grouped into sections displayed in the Navigator Panel. Groups appearing by default include Elements, Event Frames, Library, and Unit of Measure, and Analyses. When the PI Notifications feature is installed, MyPI, Notifications, and Contacts also appear in the Navigator Panel.

Browser

Use the Browser to select the objects you want to work on and display in the Viewer panel. The Browser displays the PI System objects that have been added to the AF database, such as elements, templates, notifications, etc. Depending on the section selected from the Navigator Panel, the following will be available from the Browser:

- **Elements**: Elements can be organized in several hierarchies. Users can drill down the element hierarchy created in an AF database.
- **Event Frames**: An event frame, as explained in more details in the following section, is any event, defined by a start time, an end time and a context. Event Frames can represent downtime events, process and environmental excursions, batch processing steps or any other events important to your organization.
- **Library**: This is a collection of objects that can be re-used throughout the AF hierarchy. Types of objects that appear in the Library include Categories, Element Templates, Enumeration Sets, Reference Types, and Tables.
- **Units of Measure (UOM)**: The UOM database provides automatic handling of simple conversions between units of measure for attributes of the same UOM class.
- **Analyses**: This section provides a summary of all analyses (e.g. calculations) configured on the current AF database. It allows you to perform administrative tasks like starting, stopping and backfilling analyses.

Status Bar

Check the status bar after clicking an item in the Browser to see its status. For example, last modification time, if the object is checked out or if a notification is currently loading.

Configuration Panel

This panel is used to configure properties associated with attributes such as attribute references, UOM and values for static attributes.

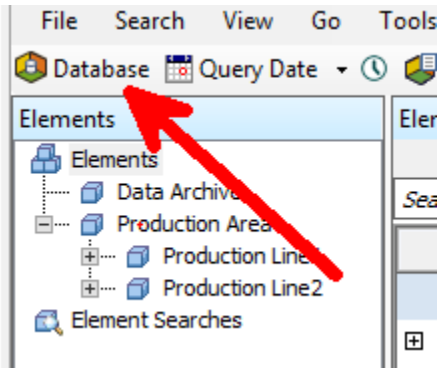
Viewer

This is the primary work area. Use it to create and edit elements, attributes, templates, tables, contacts, notifications, analyses, and so forth. When configuring attributes through the Viewer, the Configuration Panel comes into view allowing you to make configuration changes.

For more information see Using PI System Explorer (*PI System Explorer User Guide*).

Connecting to an AF Server to view the Element Hierarchy

AF stores the asset framework objects (elements, templates, and so on) in *AF databases*. You can have multiple AF databases in AF, although you can connect to only one at a time. In PSE, you can see which AF server you are connected to and its list of databases by selecting the Database button in the upper left corner.



The Select Database dialog box will appear and show you which AF server you are connected to (the drop down along the top).

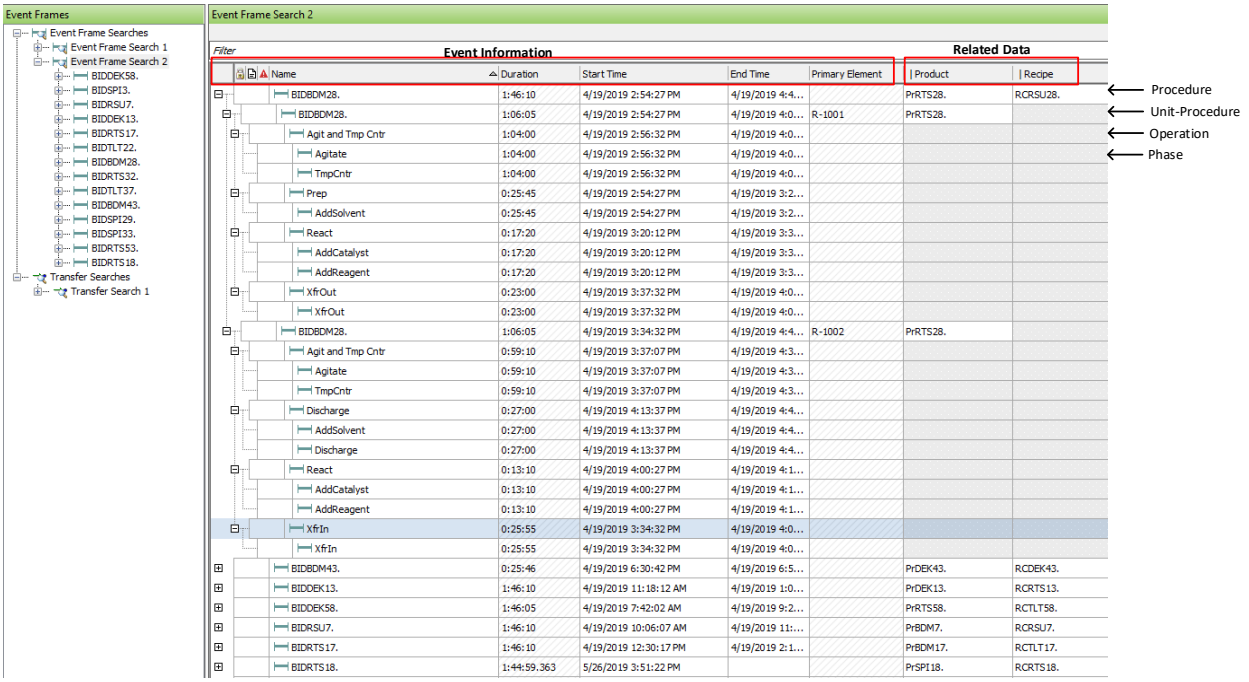
Event Frames in PI AF

Capturing important events in users' process and collecting relevant data around those events can help analyze why events occurred. For example, user can closely monitor events such as asset batch records, downtime, process excursions… to identify possible causes or potential points of failure.

Collecting data around repeatable time periods such as product tracking batches, product runs… can help make those processes more efficient. The capture of comprehensive data associated with such an event helps track, compare, or analyze the process or event.

Event Frames can be configured using PI AF Analysis, PI EF Generation Interface, or auto generate using PI Batch Interfaces.

Sample event data



For information on event frames, view videos on the EF Playlist: https://www.youtube.com/playlist?list=PLMcG1Hs2JbcunItGs7JS6kFl3WYMb-p3D&rel=0

Structure of event frames

Each EF consists of Name, Start Time, End Time, none or many attributes, and one or more referenced PI AF elements.

With EF users can easily search the PI System for events themselves, rather than search for events by time.

Event Frame Templates

PI Event Frame Generation Interface and PI Batch Interfaces automatically installed several PIBatch event frame templates, examples: PIBatch, Procedure, PIUnitBatch, UnitProcedure, PISubBatch, Operaton, Phase…

Exercise: Search Event Frames (EF)

Objectives

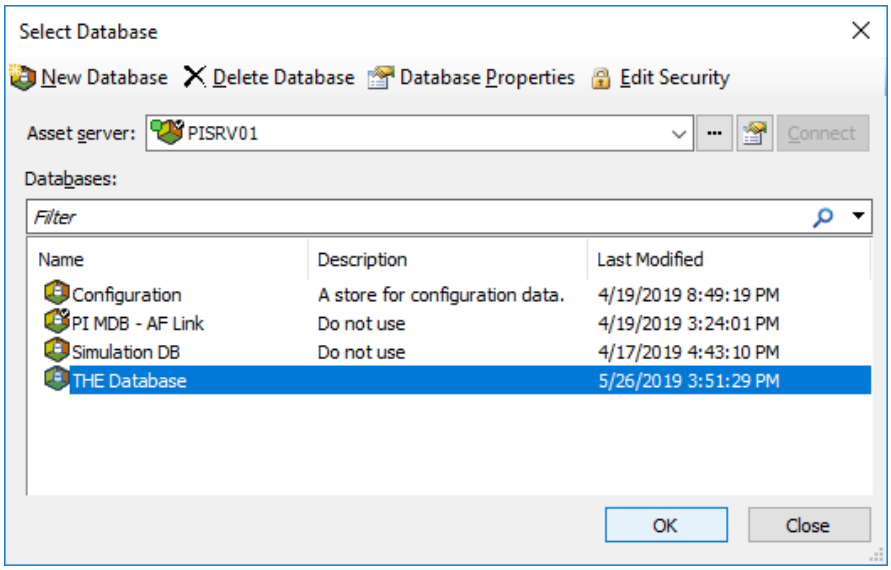
Discover PI System Explorer (PSE or PI AF Client).

Problem Description

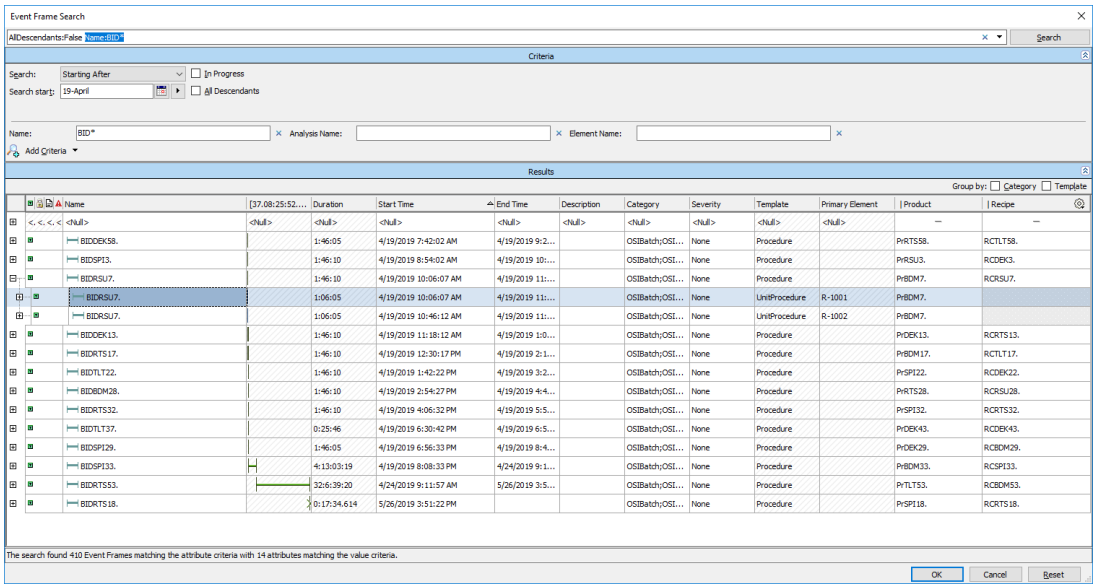
Using PSE, explore the 1K Reactors EF of Batch ID stem “BID” for the last 1 day.

Approach:

- From PIRPT01 Training Cloud Environment, open the PSE -> press Database button -> find and select “THE Database” AF database -> OK



- In the Navigation Panel -> select *Event Frames* -> right click *Event Frame Search* -> New Search
- In the Event Frame Search pop-up -> Search: Start After -> Search Start: *-1d -> Name: BID* -< uncheck *All Descendants* -> press “Search” -> OK



- In the Viewer pane, select a completed or EF (EF with start and end times) and write down the followings:
  - Procedure
    - Batch ID: _____
    - Product Name: _____
    - Recipe Name: _____
    - Start Time – End Time: _____
  - Unit-Procedure
    - Primary Element: R-1001
    - Batch ID: _____
    - Product Name: _____
    - Recipe Name: _____
    - Start Time – End Time: _____
  - Unit-Procedure
    - Primary Element: R-1002
    - Batch ID: _____
    - Product Name: _____
    - Recipe Name: _____
    - Start Time – End Time: _____

Exercise: Configure AF Elements (Assets) for Pulp and Paper Line 1

Objectives

Learn how to configure AF Elements and Attributes using PI System Explorer (PSE).

Problem Description

Configure *root element* “Pulp and Paper” production “Line 1” which consists of Chip1 and Digest1 *child elements*.

Add Attribute(s) to each *child elements*.

Approach

Open PI System Explorer

Add a *root Element* “Pulp and Paper” (Element Template: None)

Add a *child Element* “Line 1” to “Pulp and Paper” (Element Template: Line)

Add a *child PI Element* “Chip1” (*Element Template*: Chip) to Element “Line 1”

Add a *child PI Unit* “Digest1” (*Element Template*: Digest) to Element “Line 1”

Add an *Attribute* “Pres” (PI Tag: PP:Digest1.Pres; UOM: Atm) to Element “Digest1”

Add an *Attribute* “Temp” (PI Tag: PP:Digest1.Temp; UOM: C) to Element “Digest1”

Add an *Attribute* “Weight” (PI Tag: PP:Chip1.Weight) to Element “Chip1”

Exercise: Configure EF for Pulp and Paper Line 1

Objectives

Discover the PI Event Frames Interface Manager

Learn how to configure EF using PIEFGen.

Problem Description

Configure EF for Pulp and Paper Process, production Line 1 which consists of a Chip1 unit and a Digest1 unit.

Approach

Open the PI Event Frames Interface Manager

Add New Event Frame

- Configuration Name: PP:Line1
- Select event frame template: Procedure
- PIBaGen compatibility mode

PP:Line1 -> Add New Child Event Frame

- Configuration Name: UP_Chip
- Select event frame template: UnitProcedure
- Active Point
  - ✓ Tag name: PP:Chip1.AP
  - ✓ Behavior: Pulse
- Unit Procedure Data tab
  - ✓ Name new Event Frame using: PI Point value: PP:Chip1.UBID
  - ✓ Product PIPoint: PP:Chip1.Prod
  - ✓ Procedure PIPoint: PP.Chip1.Rec
  - ✓ BatchID PIPoint: PP.Chip1.UBID
- Procedure Data tab
  - ✓ PI Point value: PP:Pulp1.BID
  - ✓ Product PIPoint: PP:Pulp1.Prod
  - ✓ Recipe PIPoint: PP.Pulp1.Rec
- Reference elements tab
  - ✓ Primary Reference elements: …\Pulp and Paper\Line 1\Chip1
  - ✓ Enable

- ✓ Evaluation after event start (seconds): 5

PP-Line1 -> Add New Child Event Frame

- Configuration Name: UP_Digest
- Select event frame template: UnitProcedure
- Active Point
  - ✓ Tag name: PP:Digest1.AP
  - ✓ Behavior: Pulse
- Unit Procedure Data tab
  - ✓ Name new Event Frame using: PI Point value: PP:Digest1.UBID
  - ✓ Product PIPoint: PP:Digest1.Prod
  - ✓ Procedure PIPoint: PP.Digest1.Proc
  - ✓ BatchID PIPoint: PP.Digest1.UBID
- Procedure Data tab
  - ✓ PI Point value: PP:Pulp1.BID
  - ✓ Product PIPoint: PP:Pulp1.Prod
  - ✓ Recipe PIPoint: PP.Pulp1.Rec
- Reference elements tab
  - ✓ Primary Reference elements: …\Pulp and Paper\Line 1\Digest1
  - ✓ Enable
  - ✓ Evaluation after event start (seconds): 5

UP_Digest -> Add New Child Event Frame

- Configuration Name: OP_Digest
- Select event frame template: Operation
- Active Point
  - ✓ Tag name: PP:Digest1.OP
  - ✓ Behavior: Step
- Name new Event Frame using: Active point value

OP_Digest -> Add New Child Event Frame

- Configuration Name: PH_Digest
- Select event frame template: Phase
- Active Point
  - ✓ Tag name: PP:Digest1.PH
  - ✓ Behavior: Step
- Name new Event Frame using: Active point value

Exercise: Configure EF for Pulp and Paper Line 2

Problem Description

Configure EF for Pulp and Paper Process, production Line 2 which consists of a Chip2 element and a Digest2 element.

Approach

May follow steps of the Directed Exercise for Configure AF Elements for Line 1 and Directed Exercise for Configure EF for Pulp and Paper Line 1

Elements and Attributes Configuration

Element Name	Attribute Name	PI Point
Chip1	Weight	PP:Chip1.Weight
Digest1	Pres	PP:Digest1.Pres
	Temp	PP:Digest1.Temp
Chip2	Weight	PP:Chip2.Weight
Digest2	Pres	PP:Digest2.Pres
	Temp	PP:Digest2.Temp

Event Frame Configuration

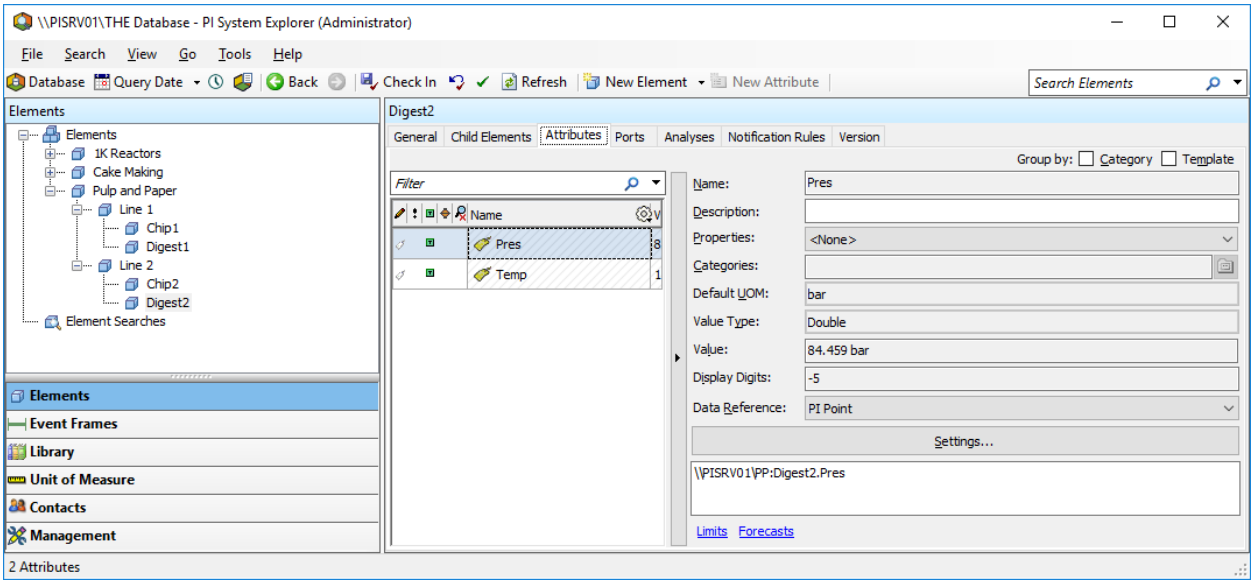
Unit	Batch Level	Name	Active Point	Batch ID Point Product Point Procedure Point
Line 1	Procedure			PP:Pulp1.BID PP:Pulp1.Prod PP:Pulp1.Rec
Chip1	Unit Procedure		PP:Chip1.AP	PP:Chip1.UBID PP:Chip1.Prod PP:Chip1.Proc
Digest1	Unit Procedure		PP:Digest1.AP	PP:Digest1.UBID PP:Digest1.Prod PP:Digest1.Proc
	Operation	OP	PP:Digest1.OP (Step)	
	Phase	PH	PP:Digest1.PH (Step)	
Line 2	Procedure			PP:Pulp2.BID PP:Pulp2.Prod PP:Pulp2.Rec

Chip2	Unit Procedure		PP:Chip2.AP	PP:Chip2.UBID PP:Chip2.Prod PP:Chip2.Proc
Digest2	Unit Procedure		PP:Digest2.AP	PP:Digest2.UBID PP:Digest2.Prod PP:Diges21.Proc
	Operation	OP	PP:Digest2.OP (Step)	
	Phase	PH	PP:Digest2.PH (Step)	

Solution: Configure EF for Pulp and Paper Line 2

Create new AF Elements

1. From PIRPT01 Training Cloud Environment, open PI System Explorer -> Elements the Navigation Panel
2. In the Browser, find and right click the element “Pulp and Paper” -> New Child Element -> Element Template: None -> Name: Line 2
3. Right click element “Line 2” -> New Child Element -> Element Template: Chip -> Name: Chip2
4. Right click element “Line 2” -> New Child Element -> Element Template: Digest -> Name: Digest2



Configure new EF

1. Open PI Event Frames Interface Manager. **Note:** always “Save Settings”
2. Interface Selection tab, Interface: PIEFGEN1
3. Event Frames Structures tab
    - Right click EFConfig_Tree -> Add New Event Frame
        - Configuration Name: PP:Line2
        - Check “PIBaGen compatibility mode
    - Right click PP:Line2 -> Add New Child Event Frame
        - Configuration Name: UP_Chip
        - Tag name: PP:Chip2.AP
        - Behavior: Pulse
        - Name New Event Frames using: PI Point value: PP:Chip2.UBID
        - Unit Procedure Data tab
            - Product: *(PIPoint):* PP:Chip2.Prod
            - Procedure: *(PIPoint):* PP:Chip2.Proc

        - BatchID: *(PIPoint):* PP:Chip2.UBID
    - Procedure Data tab
        - Name new Event Frames using: PI Point value: PP:Pulp2.BID
        - Product: *(PIPoint)* PP:Pulp2.Prod
        - Recipe: *(PIPoint)* PP:Pulp2.Rec
    - Reference elements tab
        - + -> check “Search Sub-Elements” -> Search -> Chip2
        - Check Delay evaluation of PI Points after event start -> Evaluate after event start (seconds) 5
- Right click PP:Line2 -> Add New Child Event Frame
    - Configuration Name: UP_Digest
    - Tag name: PP:Digest2.AP
    - Behavior: Pulse
    - Name New Event Frames using: PI Point value: PP:Digest2.UBID
    - Unit Procedure Data tab
        - Product: *(PIPoint):* PP:Digest2.Prod
        - Procedure: *(PIPoint):* PP:Digest2.Proc
        - BatchID: *(PIPoint):* PP:Digest2.UBID
    - Procedure Data tab
        - Name new Event Frames using: PI Point value: PP:Pulp2.BID
        - Product: *(PIPoint)* PP:Pulp2.Prod
        - Recipe: *(PIPoint)* PP:Pulp2.Rec
    - Reference elements tab
        - + -> check “Search Sub-Elements” -> Search -> Digest2
        - Check Delay evaluation of PI Points after event start -> Evaluate after event start (seconds) 5
- Right click Digest -> Add New Child Event Frame
    - Configuration Name: OP_Digest
    - Tag name: PP:Digest2.OP
    - Behavior: Step
    - Name New Event Frames using: Active point value
- Right click OP -> Add New Child Event Frame
    - Configuration Name: PH_Digest
    - Tag name: PP:Digest2.PH
    - Behavior: Step

Exercise: Configure EF for Cake Making

Problem Description

Configure EF for Cake Making Process which consists of a Cake element and a Crust element.

Approach

Create root element “Cake Making” (Element Template: None)

Create “Cake Making” child element “Cake” (Element Template: Cake)

Create “Cake Making” child element “Crust” (Element Template: Crust)

See steps in Directed Exercise – Configure EF for Pulp and Paper Line 1

Elements and Attributes

Element Name	Attribute Name	PI Point
Cake	Cheese Amount	CCF:Cake.Charge.Cheese.Qty
	Eggs Amount	CCF:Cake.Charge.Eggs.Qty
	Flour Amount	CCF:Cake.Charge.Flour.Qty
	Liquid Amount	CCF:Cake.Charge.Liquid.Qty
	Sugar Amount	CCF:Cake.Charge.Sugar.Qty
	Vanilla Amount	CCF:Cake.Charge.Vanilla.Qty
	Yolks Amount	CCF:Cake.Charge.Yolks.Qty
Crust	Butter Amount	CCF:Crust.Charge.Butter.Qty
	Cinnamon Amount	CCF:Crust.Charge.Cinnamon.Qty
	Crackers Amount	CCF:Crust.Charge.Crackers.Qty
	Nuts Amount	CCF:Crust.Charge.Nuts.Qty

Event Frame Configuration

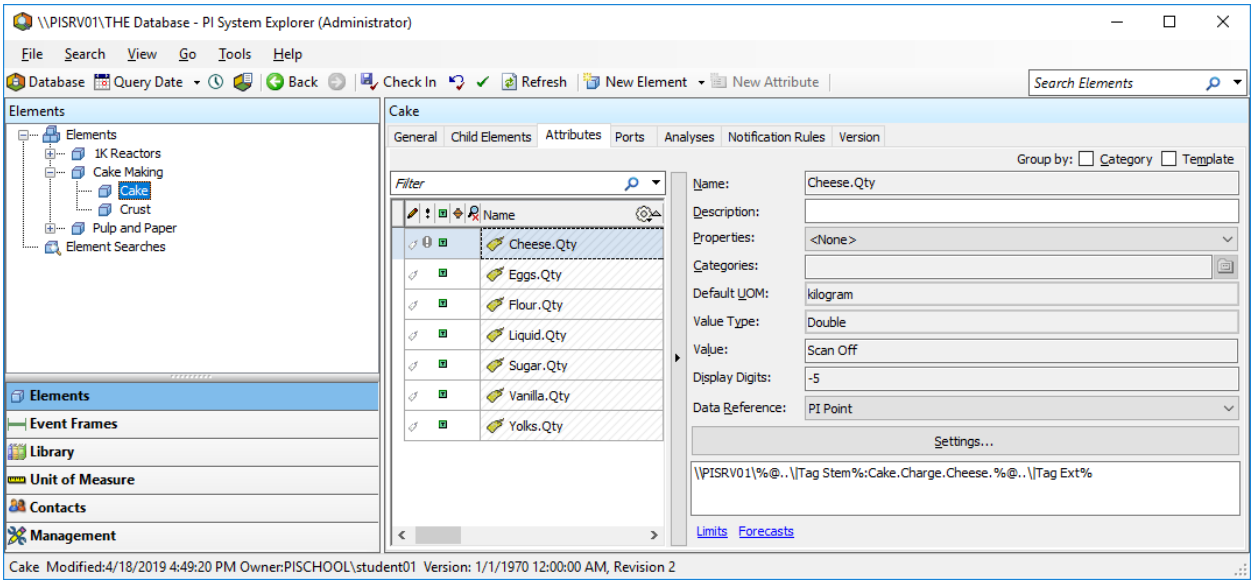
Unit	Batch Level	Name	Active Point	Batch ID Point Product Point Procedure Point
	Procedure			CCF:BID
Cake	Unit Procedure		CCF:Cake.AP	CCF:Cake.UBID CCF:Cake.Prod CCF”Cake.Proc
	Operation	Charge	CCF:Cake.Charge	
	Phase	Cheese	CCF:Cake.Charge.Cheese	
		Eggs	CCF:Cake.Charge.Eggs	
		Flour	CCF:Cake.Charge.Flour	
		Liquid	CCF:Cake.Charge.Liquid	
		Sugar	CCF:Cake.Charge.Sugar	
		Vanilla	CCF:Cake.Charge.Vanilla	
		Yolks	CCF:Cake.Charge.Yolks	

	Operation	Mix	CCF:Cake.Mix.Operation	
	Phase	Mix (step)	CCF:Cake.Mix.Phase	
	Operation	Discharge	CCF:Cake.Discharge	
Crust	Unit Procedure		CCF:Crust.AP	CCF:Crust.UBID CCF:Crut.Prod CCF:Crust.Proc
	Operation	Charge	CCF:Crust.Charge	
	Phase	Butter	CCF:Crust.Charge.Butter	
		Cinnamon	CCF:Crust.Charge.Cinnamon	
		Crackers	CCF:Crust.Charge.Crackers	
		Nuts	CCF:Crust.Charge.Nuts	
	Operation	Mix	CCF:Crust.Mix	
	Operation	Discharge	CCF:Crust.Discharge	

Solution: Configure EF for Cake Making

Consult Exercise – Configure EF for Pulp and Paper Line 2 steps.

Create new AF Elements (Assets)



Configure EF

1. Open PI Event Frames Interface Manager
2. Interface Selection tab -> Interface: PIEFGen1
3. Event Frames Structure tab
4. Configure Procedure “Cake Making”
    - Configure Unit-Procedure “Cake”
        - Configure Operation “Charge”
            - Configure Phases “Cheese”, “Eggs”, Flour, Liquid, Sugar, Vanilla, and Yolks
        - Configure Operation “Mix”
            - Configure Phase “Mix” ***(Step)***
        - Configure Operation “Discharge”
    - Configure Unit-Procedure “Crust”
        - Configure Operation “Charge”
            - Configure Phases “Cheese”, “Eggs”, Flour, Liquid, Sugar, Vanilla, and Nuts
        - Configure Operation “Mix”
        - Configure Operation “Discharge”

SAVE!!!