



## ***RtReports Administration***

***Version 4.1 Patch 4***

© 2021 AVEVA Group plc and its subsidiaries. All rights reserved.

AVEVA, the AVEVA logos and AVEVA product names are trademarks or registered trademarks of aveva group plc or its subsidiaries in the United Kingdom and other countries. Other brands and products names are the trademarks of their respective companies.

AVEVA Group plc  
High Cross, Madingley Road  
Cambridge CB3 0HB, UK  
Tel +44 (0)1223 556655  
Fax +44 (0)1223 556666

---

**aveva.com**

# Table of contents

<b>1.</b>	<b>WELCOME.....</b>	<b>6</b>
1.1	TRAINING ENVIRONMENT.....	7
<b>2.</b>	<b>BATCH ARCHITECTURE IN THE PI SYSTEM.....</b>	<b>8</b>
2.1	BACKGROUND: INTRODUCTION TO ISA S88.....	8
2.2	BACKGROUND: EXAMPLES OF BATCH PROCESSES .....	9
<b>3.</b>	<b>RTREPORTS OVERVIEW.....</b>	<b>10</b>
3.1	BACKGROUND: RTREPORTS COMPONENTS .....	10
3.2	BACKGROUND: WHY USE RTREPORTS? .....	11
3.3	BACKGROUND: RTREPORTS COMPLIANCE WORKFLOW.....	12
3.4	BACKGROUND: RTREPORTS CONTEXTS.....	14
3.4.1	Batch Report Template.....	14
3.4.2	Time Report Template .....	14
3.5	BACKGROUND: AUTOMATIC REPORT GENERATION.....	14
<b>4.</b>	<b>CONFIGURE BATCH REPORTS.....</b>	<b>15</b>
4.1	BACKGROUND: BATCH REPORT TEMPLATES.....	15
4.2	BACKGROUND: DATA TEMPLATE .....	15
4.2.1	Extract the Batch Hierarchy .....	16
4.2.2	Apply the Business Rules .....	17
4.3	BACKGROUND: UNIT CLASS .....	17
4.3.1	Configure Unit Class.....	17
4.4	KNOWLEDGE CHECK: CONFIGURE UNIT CLASS.....	18
4.5	BACKGROUND: BATCH JOURNAL.....	19
4.5.1	Properties, Columns, and Events.....	19
4.5.2	Walking Up the Journal .....	19
4.5.3	Terminal States vs. Sequence Entries .....	20
4.6	BACKGROUND: BATCH REPORT TEMPLATE WIZARD.....	20
4.7	HANDS-ON: CREATE A BATCH REPORT TEMPLATE.....	21

<b>5.</b>	<b>DATA FORMATTING .....</b>	<b>25</b>
5.1	BACKGROUND: DEFAULT TEMPLATE .....	25
5.2	BACKGROUND: FORMATTING ELEMENTS .....	25
5.3	BACKGROUND: HIERARCHIES .....	26
5.4	BACKGROUND: FORMATTING .....	26
5.5	HANDS-ON: FORMAT A REPORT .....	28
<b>6.</b>	<b>USING ITERATIONS .....</b>	<b>34</b>
6.1	BACKGROUND: WORKING WITH ITERATOR EXPRESSIONS .....	34
6.2	BACKGROUND: ITERATOR EXPRESSION ENTRY .....	34
6.3	BACKGROUND: MULTI-LEVEL ITERATORS .....	35
6.4	HANDS-ON: USE ITERATORS .....	36
<b>7.</b>	<b>JOURNAL ACTIONS .....</b>	<b>40</b>
7.1	BACKGROUND: CONTEXT .....	40
7.2	BACKGROUND: ACTION PARAMETERS AND RESULTS .....	40
7.2.1	Action Parameters .....	40
7.2.2	Trend Result Definition .....	41
7.2.3	TraceSet Result Definition Types .....	41
7.3	HANDS-ON: CONFIGURE A JOURNAL ACTION .....	42
7.4	HANDS-ON: DISPLAY COMPRESSED VALUES .....	48
<b>8.</b>	<b>BUILDING TIME REPORTS .....</b>	<b>52</b>
8.1	BACKGROUND: TIME REPORT TEMPLATES .....	52
8.1.1	Recurrence .....	52
8.1.2	Reference Start .....	52
8.1.3	Report Duration .....	52
8.1.4	Applying Time Rules .....	53
8.1.5	Modify Time Rules .....	53
8.2	BACKGROUND: TIME BASED ROOT JOURNAL .....	53
8.3	BACKGROUND: USING EQUIPMENT IN A TIME REPORT .....	53

<b>8.4</b>	<b>HANDS-ON: CONFIGURE A TIME REPORT TEMPLATE .....</b>	<b>54</b>
<b>8.5</b>	<b>KNOWLEDGE CHECK: CONFIGURE A DAILY SHIFT TIME REPORT.....</b>	<b>64</b>
<b>9.</b>	<b>CONDITION BASED REPORTING .....</b>	<b>69</b>
<b>9.1</b>	<b>BACKGROUND: CONFIGURE ACTION STATES (DATA TEMPLATE) .....</b>	<b>69</b>
<b>9.1.1</b>	<b>Timing of the Action State Transitions Evaluation .....</b>	<b>70</b>
<b>9.1.2</b>	<b>Filter or Suppress Transitions Evaluations .....</b>	<b>71</b>
<b>9.1.3</b>	<b>More Action Options .....</b>	<b>71</b>
<b>9.2</b>	<b>HANDS-ON: REPORT BY EXCEPTION WITH ACTION STATES .....</b>	<b>72</b>
<b>9.3</b>	<b>BACKGROUND: APPLY FORMATTING CRITERIA .....</b>	<b>79</b>
<b>9.3.1</b>	<b>Format Template Condition Field .....</b>	<b>79</b>
<b>9.3.2</b>	<b>Iterators: Iterator Result Condition .....</b>	<b>79</b>
<b>9.3.3</b>	<b>XPath Functions .....</b>	<b>80</b>
<b>9.4</b>	<b>KNOWLEDGE-CHECK: CONDITIONAL FORMATTING.....</b>	<b>81</b>
<b>10.</b>	<b>(OPTIONAL) ADVANCED TOPICS FOR RTREPORTS EDITOR.....</b>	<b>84</b>
<b>10.1</b>	<b>BACKGROUND: FORMAT TEMPLATE OPTIONS .....</b>	<b>84</b>
<b>10.1.1</b>	<b>Disable Comments and Verifications .....</b>	<b>84</b>
<b>10.1.2</b>	<b>Printing .....</b>	<b>84</b>
<b>10.1.3</b>	<b>Security, Reviews and Approvals.....</b>	<b>85</b>
<b>10.2</b>	<b>BACKGROUND: FINISHING A REPORT TEMPLATE.....</b>	<b>86</b>
<b>10.2.1</b>	<b>Report Template Status.....</b>	<b>86</b>
<b>10.2.2</b>	<b>Releasing a Report Template.....</b>	<b>86</b>
<b>10.2.3</b>	<b>New Version for a Report Template.....</b>	<b>87</b>
<b>10.3</b>	<b>BACKGROUND: IMPORTING AND EXPORTING.....</b>	<b>87</b>
<b>10.3.1</b>	<b>Import/Export a Report Template .....</b>	<b>87</b>
<b>10.4</b>	<b>KNOWLEDGE CHECK: IMPORT AND RELEASE A BATCH REPORT TEMPLATE..</b>	<b>89</b>
<b>10.5</b>	<b>KNOWLEDGE CHECK: IMPORT AND RELEASE A TIME REPORT TEMPLATE.....</b>	<b>91</b>
<b>10.6</b>	<b>BACKGROUND: REPORT PARAMETERS .....</b>	<b>94</b>
<b>10.7</b>	<b>KNOWLEDGE CHECK: CREATE AND RELEASE A REPORT WITH PARAMETERS</b>	
	<b>95</b>	
<b>11.</b>	<b>(OPTIONAL) PI BASELINE SERVICE .....</b>	<b>99</b>
<b>11.1</b>	<b>BACKGROUND: INTRODUCTION .....</b>	<b>99</b>

<b>11.2</b>	<b>BACKGROUND: DATA SOURCES</b> .....	<b>100</b>
<b>11.3</b>	<b>BACKGROUND: DATASETS</b> .....	<b>101</b>
<b>11.4</b>	<b>KNOWLEDGE CHECK: INCORPORATE EXTERNAL DATA IN REPORT</b> .....	<b>102</b>
<b>12.</b>	<b>(OPTIONAL) RTREPORTS SECURITY</b> .....	<b>106</b>
<b>12.1</b>	<b>BACKGROUND: INTRODUCTION</b> .....	<b>106</b>
<b>12.2</b>	<b>BACKGROUND: RTREPORTS WINDOWS GROUPS</b> .....	<b>106</b>
<b>12.3</b>	<b>BACKGROUND: GRANT PERMISSIONS TO GROUPS</b> .....	<b>107</b>
<b>12.4</b>	<b>BACKGROUND: REPORT TEMPLATE SECURITY</b> .....	<b>108</b>
<b>12.4.1</b>	<b>Report Edit and Report Execute</b> .....	<b>108</b>
<b>12.4.2</b>	<b>Grant RtReports Windows Groups to a Report Template</b> .....	<b>108</b>
<b>12.5</b>	<b>KNOWLEDGE CHECK: GROUP AND USER SECURITY</b> .....	<b>109</b>
<b>13.</b>	<b>(OPTIONAL) RTREPORTS INSTALLATION</b> .....	<b>116</b>
<b>13.1</b>	<b>BACKGROUND: RTREPORTS SERVER SETTINGS</b> .....	<b>118</b>
<b>13.2</b>	<b>BACKGROUND: RTREPORTS CACHE MANAGEMENT SETTINGS</b> .....	<b>119</b>
<b>13.3</b>	<b>BACKGROUND: RTREPORTS EDITOR STANDALONE INSTALLATION</b> .....	<b>119</b>
<b>14.</b>	<b>FINAL EXAM</b> .....	<b>120</b>

# 1. Welcome



Before reading this section, please refer to the following course YouTube video:  
[RtReports Administration - Course Introduction](#)

Welcome to the RtReports Administration Course.

This training has been crafted to equip you with the skills needed to design web-based, regulatory-compliant reports using the RtReports applications. Therefore, if you are a PI System Manager or an advanced user who is responsible for the administration of RtReports at your organization, this training is an excellent choice for you.

Throughout this course, we will guide you in acquiring comprehensive knowledge about the RtReports Editor component. You will learn how to create various types of reports, such as PI Batch-Based reports or Time-Based reports.

Additionally, you'll discover how to select data sources, how to format your report to ensure it meets your specific requirements, and how to filter out unnecessary data to ensure key information isn't missed.

As you progress, you'll work with your own training cloud environment allowing you direct testing and application of new concepts.

Throughout this class we will assume you have a basic knowledge of the PI System. If you are new to the PI System, we encourage you to complete the [PI System Basics](#) class that is available for free on our Learning Page.

The course consists of text-based lessons, and video lectures. You are encouraged to follow along with all video lectures using your Training Cloud Environment. Text-based lessons contain background theory, exercises, documentation, and other useful information.

Lessons are categorized using the following key words:

1. **Hands-On:** These lessons contain hands-on exercises with video step-by-step walkthroughs. While we are demonstrating how to use different functions of RtReports Editor, you are encouraged to do the same on your virtual environment.
2. **Knowledge check:** Exercises with step-by-step solutions to complete on your own using your virtual environment, or quizzes to keep your skills sharp.
3. **Background:** Readings or videos containing useful background information to assist your understanding. These lessons are optional but encouraged as they will give you a more well-rounded understanding of the features of RtReports. Throughout these sections we will link out

to reference documentation. You may be examined on the content inside these references, but as the exam is open book, you're welcome to visit these references during the exam.

In the next lesson we'll cover the Learning Cloud Environment used in this class.

## 1.1 Training Environment

The environment for this course is hosted in Microsoft Azure. There are 5 Virtual Machines, each consisting of the following:

- **PIDC** – Domain Controller
- **PICLIENT01** – Client Computer
  - RtReports Editor
  - PI System Management Tools
  - PI System Explorer
  - PI Event Frames Generator
  - Google Chrome
  - Outlook (Domain: PISCHOOL; Users: student01, student02, student03)
  - Folder C:\Class
- **PIRTR01** – RtReports Server
  - RtReports Server
  - PI System Explorer
  - PI System Management Tools
- **PISRV01** – PI System Server
  - Microsoft SQL Server 2019
  - PI Data Archive
  - PI AF Server
  - hMailServer

The Login Username: **PISCHOOL\student01** and password: <TBA>.

## 2. Batch Architecture in the PI System

### 2.1 Background: Introduction to ISA S88

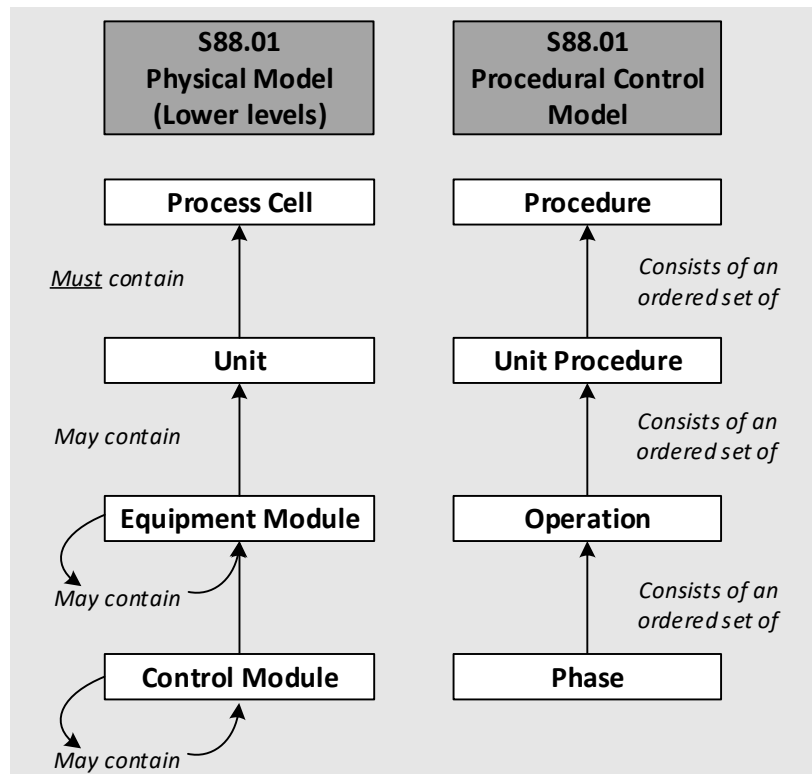
In a plant where production is performed in batches (repetitive cycles), event frames can be used to capture the production batches in AF Event Frames.

A batch process could be seen in a pharma company where batches of a particular medicine are produced in a reactor. Every batch consists of multiple steps such as e.g.: filling components into a reactor, mixing the components, heating up to start a chemical process, emptying the resultant liquid from the reactor and finally bottling. There is a defined process of steps (called the “recipe”) that happens repeatedly through time.

ISA (International Society of Automation) has published the ISA-S88.01 standard under the name “Batch Control: Models and Terminology” to supply an industry standard for naming or referring to stages in batch processes. Its original version was adopted by the IEC in 1997 as IEC 61512-1. The standard provides better communication between vendors and buyers of process control systems.



Two models defined by the S88 standard are commonly followed when structuring information: the **Physical Model**, a hierarchy used to describe the enterprise assets in the batch process, and the **Procedural Control Model**, characterizing the actions and events taking place sequentially for the batch process.



Comparison of ISA S88 Physical and Procedural Control models

In modern PI systems, the ISA S88 models can be applied directly to the PI Asset Framework: for example, the physical assets can be defined as AF Elements, and the procedural control hierarchy can be



designed as AF Event Frames (EF). Further discussion on AF usage for batch processes is in the next chapters.

Note that AF hierarchies do not have to strictly adhere to specific mappings of ISA S88 – there is a lot more flexibility to model structures differently to whatever is desired for business operations.

In legacy PI Data Archive environments, the Procedural Control Model may be applied on the PI Batch Database (BDB), and the Physical Model can be applied to the PI Module Database (MDB).

## 2.2 Background: Examples of Batch Processes

Batches 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 Batch applications:

---

---

## 3. RtReports Overview



Before reading this section, please refer to the following course YouTube video:  
[RtReports – What is it?](#)

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 PI Baseline 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.

### 3.1 Background: RtReports Components

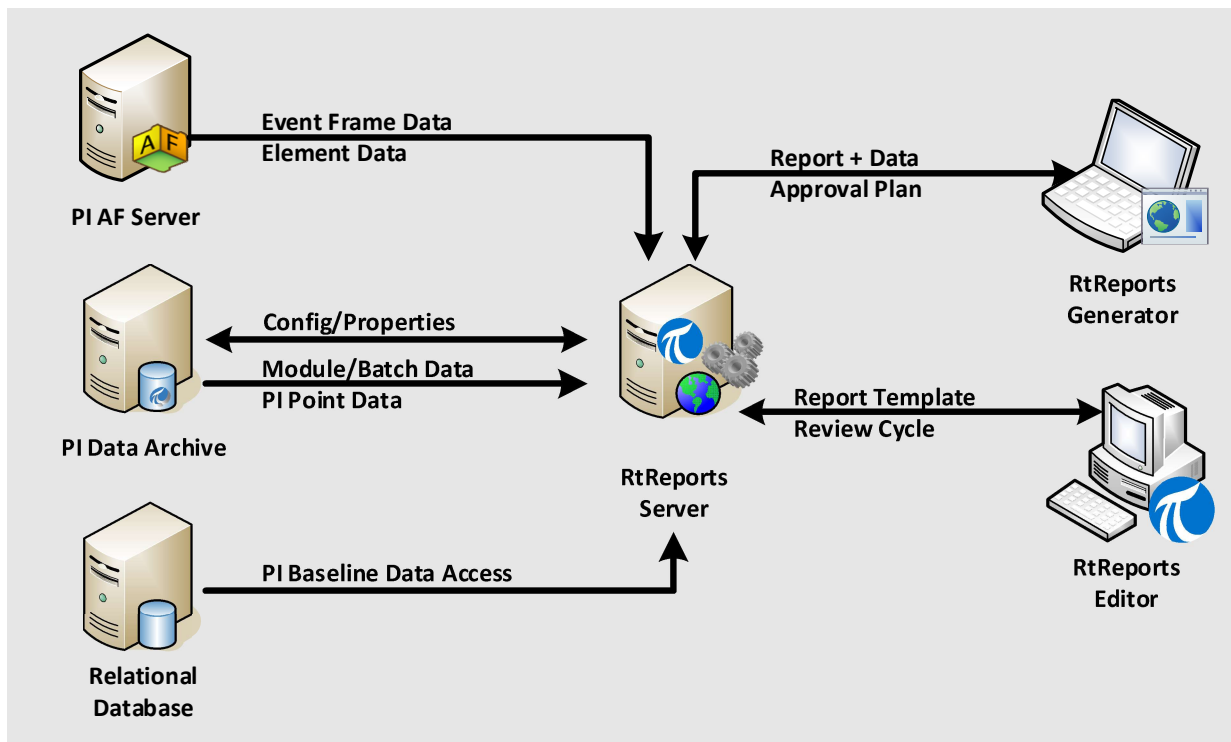
RtReports is not just one single entity. It is made up of a series of components that all work in harmony.

**RtReports Editor** – a .NET Windows Application that allows users to design and test Report Templates against multiple sources of PI System data, using rules and conditions to format, filter out, and combine values for any repeatable context, based on batches or time frames. The Editor also includes administrative functions such as tuning user access to reports and a built-in versioning system: the review of a report template must be electronically signed for release.

**RtReports Generator** is the .NET web Application (Web Browser URL: **Error! Hyperlink reference not valid.**), allows users to search data context within PI AF Server and PI Data Archive server, aggregate data according to the specified report template rules, identify deviations from critical process limits, maintain comments and approval history, and generate pre-configured reports.

**RtReports Server** – hosted on an Internet Information Services (IIS) Server and based on ASP.NET, it serves the data requested by the Editor and Generator applications. The PI Baseline service is also hosted there, allowing users to configure and aggregate data from an external Relational Database server.

The RtReports Server audits the actions triggered from Editor and Generator applications, and all information is stored in the default PI Data Archive. The officially approved reports are generated as secured Portable Document Format (PDF) files, available to download by the end users.



**RtReports Architecture**

## 3.2 Background: 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.

### **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. The PI System core infrastructure 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.

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

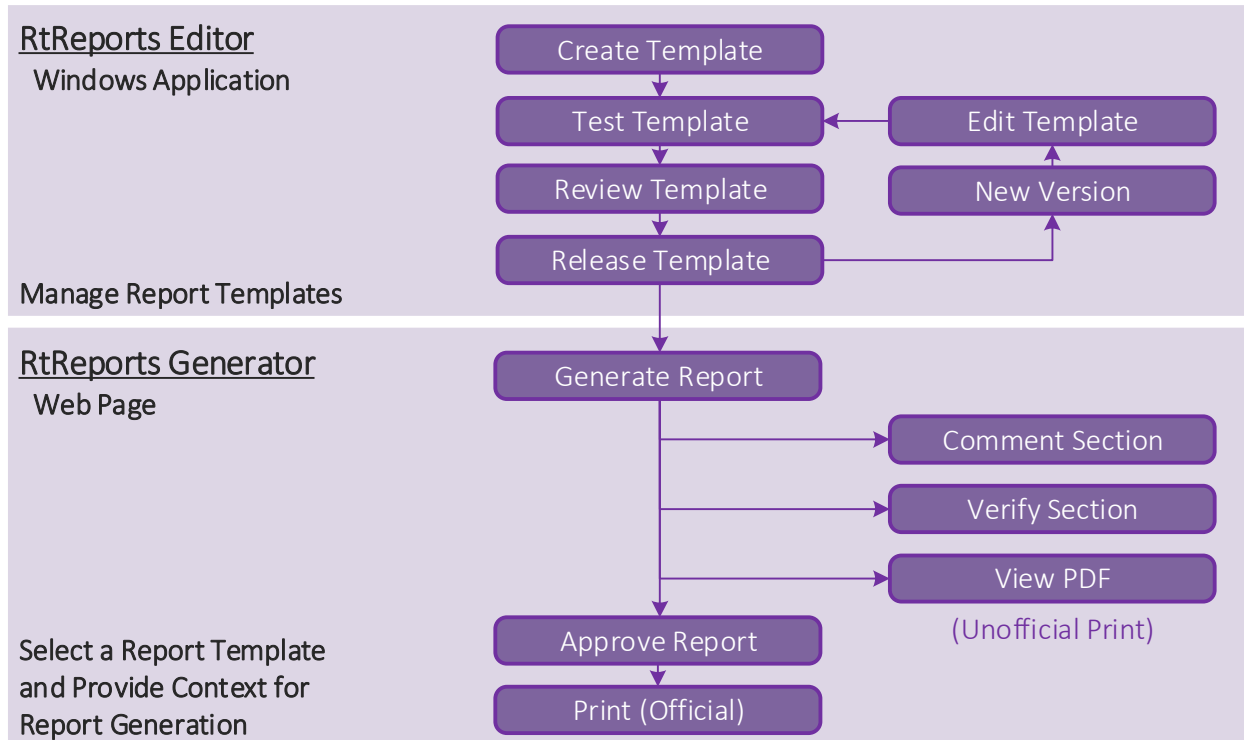
### **Wasatch Energy Systems Case Study**

Wasatch Energy Systems of Layton, Utah, uses the PI System to meet regulatory requirements for monitoring, recording, and reporting environmental data related to NO<sub>x</sub>, SO<sub>x</sub>, 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 will learn how PI addresses system availability, data reporting, and auditing along with details of project execution.

## **3.3 Background: RtReports Compliance Workflow**

The RtReports workflow has many benefits, including:

- Security for Report Template configuration
- Security for Reviewing
- Security for Report commenting, verifying, printing, and report approval
- All inputs require electronic signature



## RtReports Compliance Workflow

### Report Template Status

When a Report Template is created, it is created with a status of “In Edit”. 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”.

## **3.4 Background: RtReports Contexts**

### **3.4.1 Batch Report Template**

The Batch Report Templates allow users to configure Report Templates that can be generated for a specific Procedure (that is a PI AF Event Frame or a PI Batch) from a PI Asset Framework Database or a legacy PI Batch Database. Each Procedure includes all the sub-items such as the Unit Procedures, Operations, and Phases.

### **3.4.2 Time Report Template**

The 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.

## **3.5 Background: Automatic Report Generation**

The Automatic Report Generation (ARG) feature is available starting in RtReports 4.1.

The ARG is an IIS Preload service that runs under the RtReports application pool on the web server. It allows users to enable auto report generation for released Batch or Time report based on users' specific criteria(s), such as Event name, PIBatch Product, PIBatch Recipe

## 4. Configure Batch Reports

RtReports allow users to configure Report Templates that can be generated for specific Event Frames from a PI Asset Framework (AF) Database – in legacy environments, a specific PI Batch in the PI Batch Database). It can also include configuration the sub-items, typically, Unit Procedures, Operations and Phases (or PI UnitBatches and PI SubBatches). Those events are often populated by PI Batch Interfaces, which read data from an external source or from PI Points, like the PI Event Frames Generator Interface.

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.

### 4.1 Background: Batch Report Templates

Report Template criteria can define the relationship between the desired batch database and the Batch Report Template. Use the criteria settings to associate a Report Template to a named Recipe, Product, Equipment, SubBatch, Unit Class, or any combination.

The second step of building a Report Template consist in defining the business rules that will be used to retrieve data from PI Servers (or external Relational Databases). 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 as PIBatch, UnitBatches and SubBatches.

### 4.2 Background: Data Template

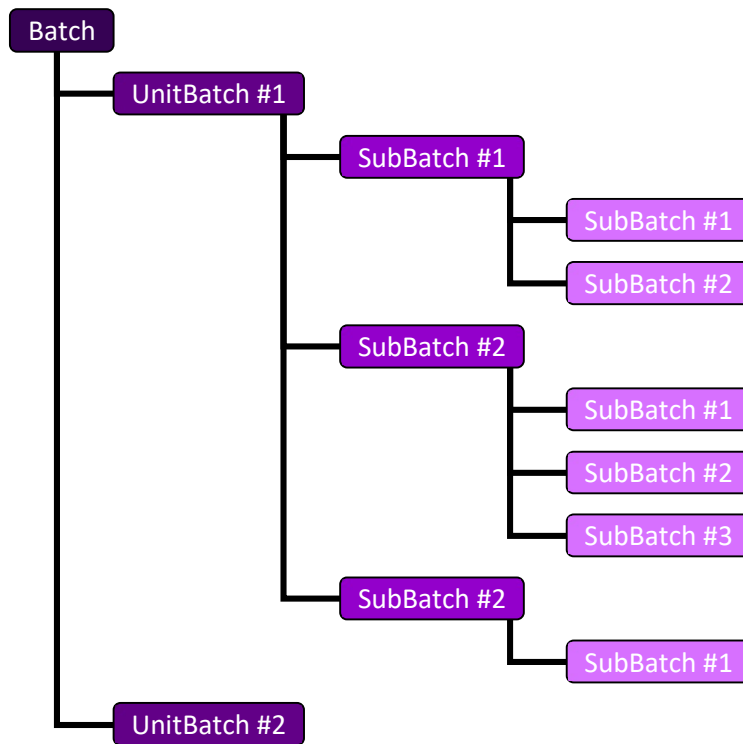
#### Defining Data Retrieval through the Data Template

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 Asset Framework, the PI Data Archive (including the legacy PI Batch Database and the PI Module Database), or an external Relational Database. Next, the rules are processed, and the “Journal” (the results set to be formatted) 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 (a top-level Event Frame), Unit Templates (an EF linked to an AF Element, or a Unit Batch linked to a PI Unit from the legacy PI MDB), and Sub-Batch Templates (child EFs like operations and phases) to describe the mapping rules to include these batch objects in the report.



For example, suppose you want to create a report for a product produced by a batch run through Units R-2401, R-2402, and R-2403. 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.

#### 4.2.1 Extract the Batch Hierarchy

Using a batch template, the report will always start at the Batch (Procedure) level. At this time, the user should decide which child batch objects he wants to see. It is possible to add a Unit Template to extract data from the Unit Procedures (PI UnitBatches) under the Batch level. In a similar way, it is also possible to add a SubBatch Template to extract data from the child Event Frames (PI SubBatches) included in the Unit Procedure or another SubBatch object, if we add it under another SubBatch Template.

In every case, it is possible to use generic extracting business rules that will apply to all child levels of the batch hierarchy. It is also possible to restrict the business rules to one or only a few Unit Procedures or specific child Event Frames levels.



### 4.2.2 Apply the Business Rules

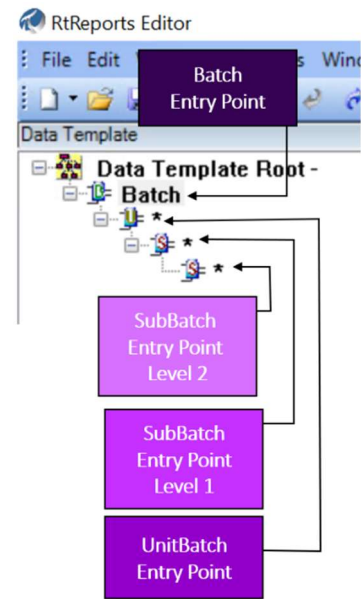
Applying the Business Rules to a context for a batch-based report means to run the rules against a specific batch. This action will have as a result the creation of a journal.

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 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.



### 4.3 Background: Unit Class

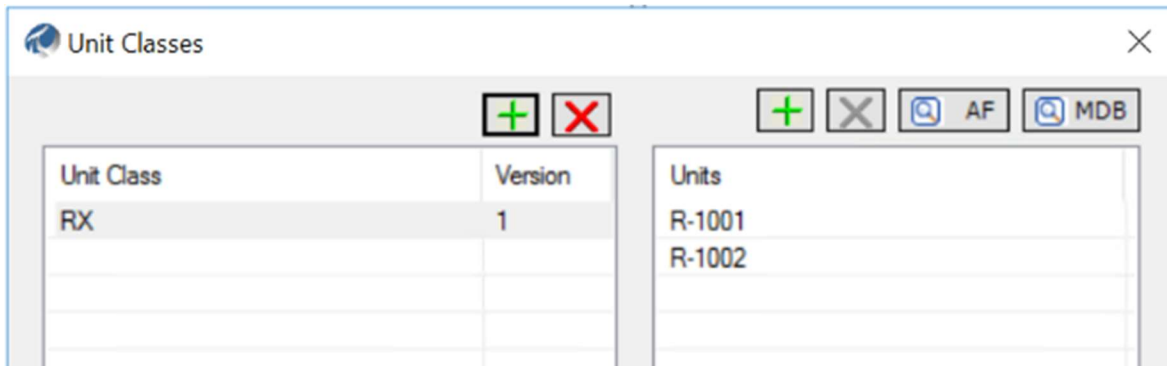
Unit Classes allow you to group equipment together and then reference the Unit Class definition. Using Unit Classes in conjunction with unit relative

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

#### 4.3.1 Configure Unit Class

Configuring a Unit Class is as simple as giving a name to the new class and then adding individual unit(s) in the 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 Unit Procedures (or PI UnitBatches) that belong to the class defined. **It then ignores the name pattern.**



## 4.4 Knowledge check: Configure Unit Class

### Objectives

- Discover the RtReports Editor
- Learn how to configure Unit Class

### Problem Description

Users would like to organize their units into classes for easier categorization.

Use the Unit Class Editor Tools to add a Unit Class called RXs and place the two reactors R-1001 and R-1002 into it.

### Solution:

1. Launch the RtReports Editor on the PIClient01 Training Cloud Environment using the Start > PI System > RtReports Editor -> and login.
2. Tools -> Unit Class Editor...
3. Click the green + above the Unit Class pane -> Unit Class Name: RXs -> Unit Class Version: 1.0 -> OK
5. Select the "RXs" in the Unit Class pane -> click the green + above Units pane -> call it "R-1001" -> repeat clicking the green + above Units pane -> call it "R-1002"
7. Click on "OK".

## 4.5 Background: Batch Journal

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

At the top level, we have the BatchJournal. This level holds all the top-level batch 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.

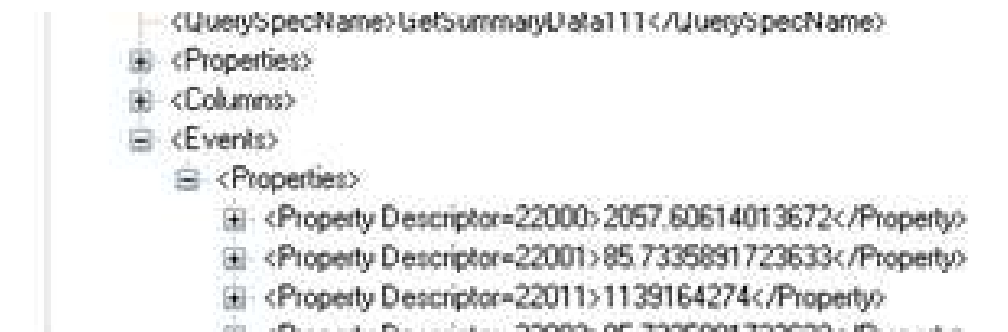
The data for the second level (linked to a unit) is located under the UnitBatchJournals which has a collection of UnitBatchJournal. Each one of these UnitBatchJournal holds the information of one specific Unit Procedure (PIUnitBatch) such as Batch ID, Product, Procedure Name, Start and End Times, Duration,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

Lower levels are stored under SubBatchJournals, which has a collection of SubBatchJournal. Each one of these SubBatchJournal holds the information of one specific Operation/Phase (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 sub-batch level.

### 4.5.1 Properties, Columns, and Events

Many 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. By combining all three pieces we get the label “Maximum” and the actual result value for that label.

### 4.5.2 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.

### 4.5.3 Terminal States vs. Sequence Entries

The data in the Terminal States branch are identical to the data in the Sequence Entries branch.

**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.

Which branch of data should be use? 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**.

## 4.6 Background: Batch Report 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.

## 4.7 Hands-on: Create a Batch Report Template



Before reading this section, please refer to the following course YouTube video:  
[RtReports Administration - Create a Batch Report Template](#)

### 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

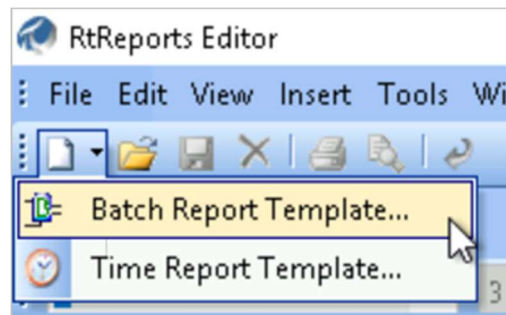
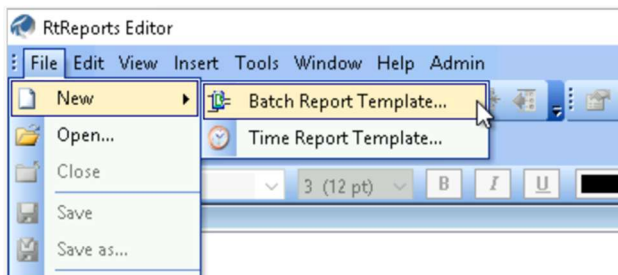
You are asked to create a report template for a batch process. This exercise will focus on the construction of the Data Template section. Your report should be built in a way that you can extract data from the PI Batch, PI UnitBatch, and PI SubBatches level.

### 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).

### Step by step solution:

1. Login to RtReports Editor: Start ► PI System ► RtReports ► RtReports Editor ► ...
2. There are two ways to do create a new report template:
  - Select File ► New ► Batch Report Template...
  - Select the New Icon ► Batch Report Template...



3. In the name field, type in: Batch\_Report

**Batch Report Template Details**

**Attributes**

Name:

Description:

**Criteria**

Recipe:

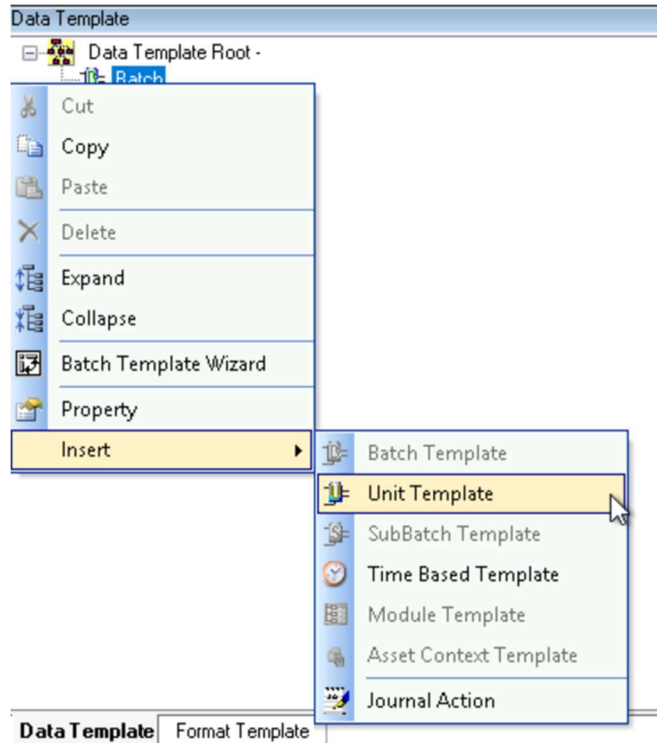
Product:

Equipment:

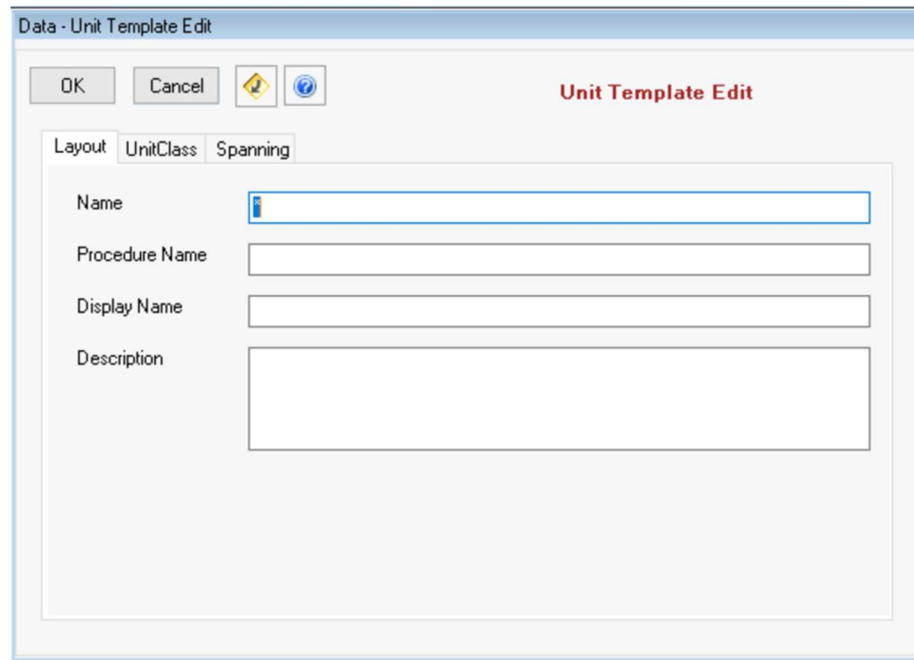
Sub Batch:

UnitClass	Version

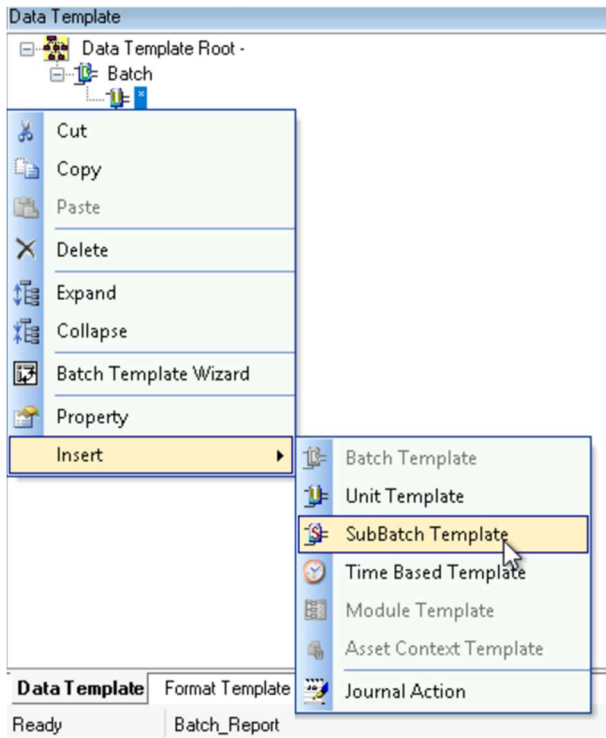
4. Leave the remaining boxes unchanged ► *OK*
5. Add a Unit Template. There are 3 ways to do so:
  - a. Right-click on *Batch* ► select *Insert* and click on *Unit Template*.
  - b. Select *Batch* by clicking on it, then click on *Insert*, select *Template*, and click on *Unit Template*.
  - c. Hover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



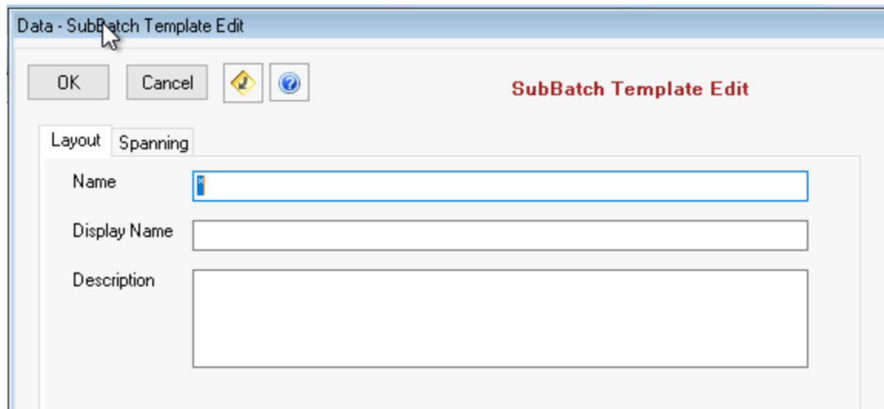
6. *Name:* \*



7. Leave the other boxes empty ► *OK*
8. 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 \*



9. *Name: \**



10. Leave the other boxes unchanged ► *OK*

11. *Repeat the steps 8-10 with the SubBatch Template selected to create a child sub-batch SAVE!!!*



## 5. 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.

### 5.1 Background: 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.

### 5.2 Background: Formatting Elements

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

#### Grouping 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 Elements

- 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

- 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.

## 5.3 Background: Hierarchies

Normally you will be held to a certain hierarchy. That is, you will first add a Section, and then an 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.

## 5.4 Background: Formatting

Almost everything that is visible can be controlled through some sort of format settings. The following items 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)

### Numeric Format

Number Format	Name	Example
C or c	Currency	C2: \$75. <u>33</u>
E or e	Scientific	E3: 7.533E+001
F or f	Fixed-point	F4: 75. <u>3315</u>
P or p	Percent	P2: 7,533. <u>15</u> %

**Date and Time Format**

<b>Number Format</b>	<b>Name</b>
yy or yyyy	Year: 2 or 4 digits
M or MM	Month: Short or Long number
MMM or MMMM	Month: Short or Long name
d or dd	Day: Short or Long number
ddd or dddd	Day: Short or Long name
h or hh – H or HH	Hour: Short or Long number (12h or 24h clock)
m or mm	Minute: Short or Long number
s or ss	Second: Short or Long number
z or zz	Time zone: Short or Long name

## 5.5 Hands-on: Format a Report



Before reading this section, please refer to the following course YouTube video:  
[RtReports Administration - Format a Report](#)

### Objectives

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

Discover RtReports Report Preview pane

### Problem Description

Configure Daily Batch Report.

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

Start from the Batch\_Report from the previous exercise

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 *PI Test Report*, using Arial font, bold formatting and 24 pt.

Insert the image RtReports\_01.gif from %PIHOME%\RtReports\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**

The screenshot displays the RtReports Editor interface. On the left, the 'Format Template' tree shows the structure of the 'Batch\_Report' template, with 'Main Batch Data' selected. On the right, the 'Report' preview shows the rendered output, including a cover sheet with the RtReports logo and the title 'PI Test Report'. Below the title, a table titled 'Main Batch Data' contains the following information:

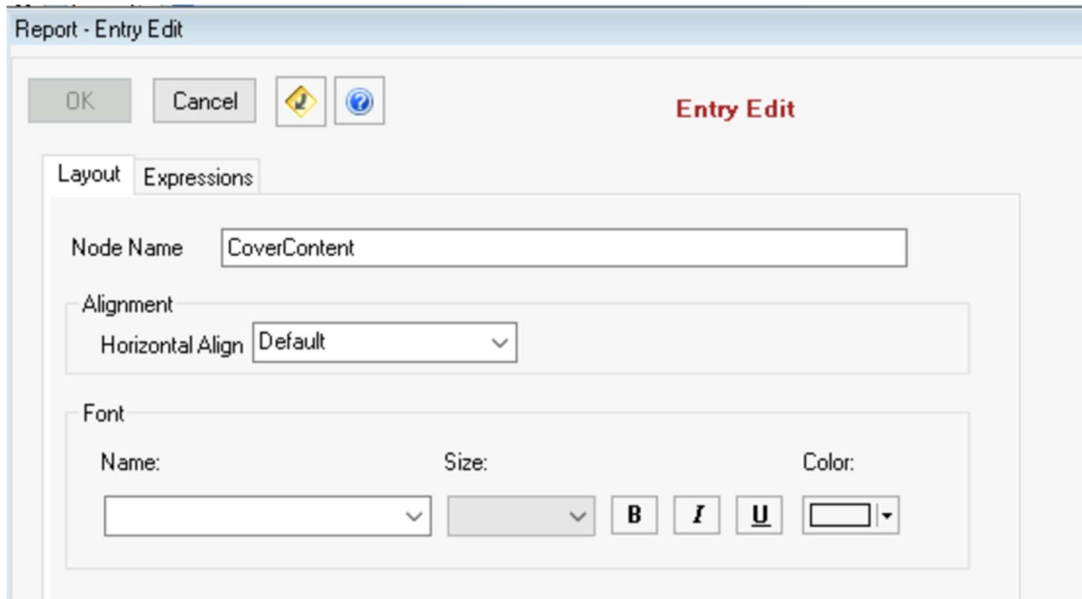
Batch ID	Start Time	End Time
BIDDEK2.	29-May-22 10:59:02 AM	29-May-22 12:33:32 PM

**Step by step solution:**

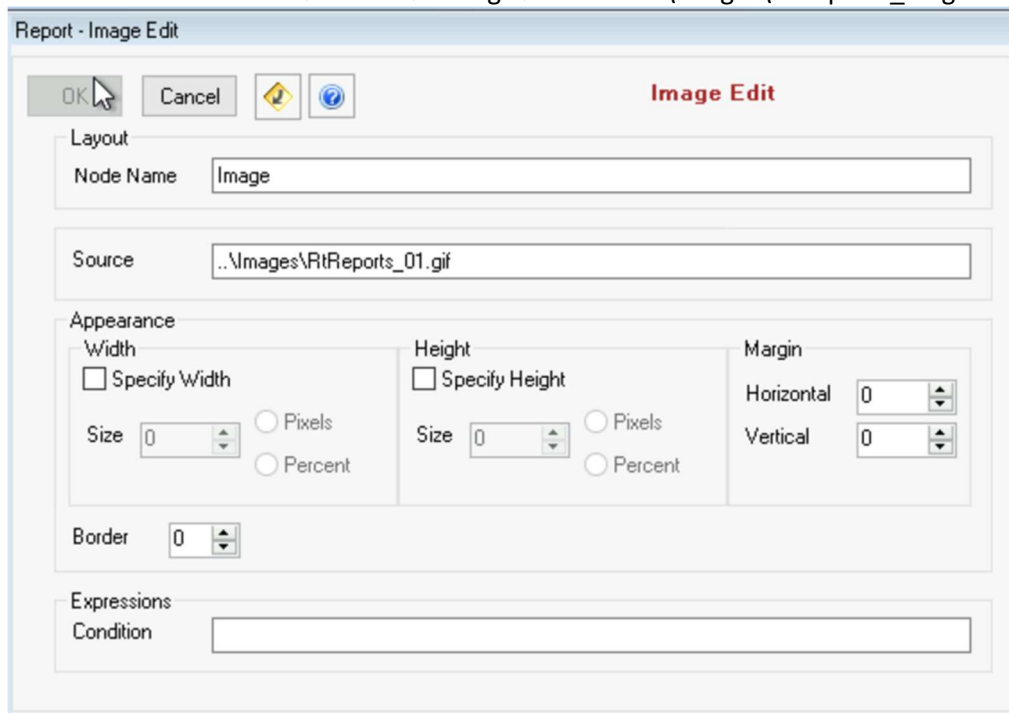
**Building Report Content**

1. Login to RtReports Editor
2. Open the "Batch\_Report" report template from the previous exercise
3. Select the *Format Template* tab
4. Select the CoverSheet ► insert ► Entry... ► Node Name: CoverContent ► OK

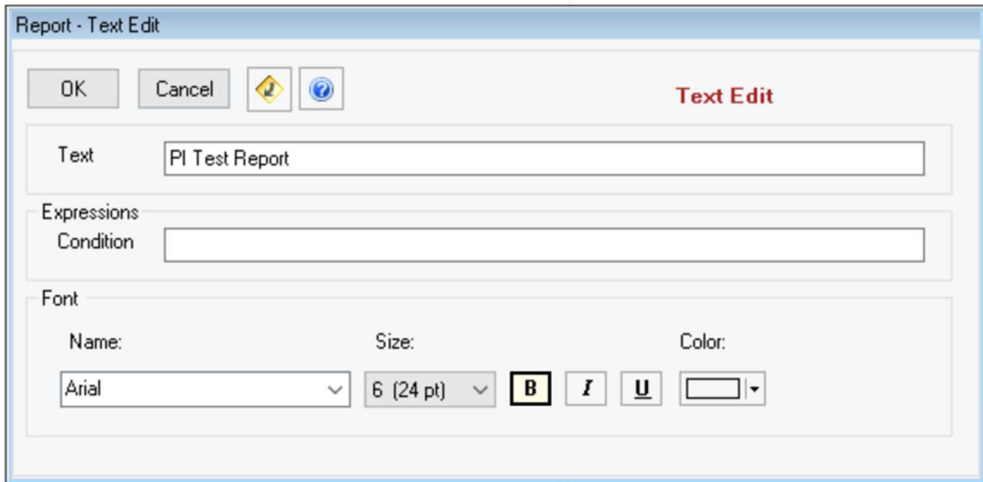
This screenshot shows the 'Format Template' tree with 'CoverSheet' selected. A context menu is open over 'CoverSheet', and the 'Insert' option is expanded to show 'Entry' selected. The 'Entry' option is highlighted, indicating it is the selected item for insertion.



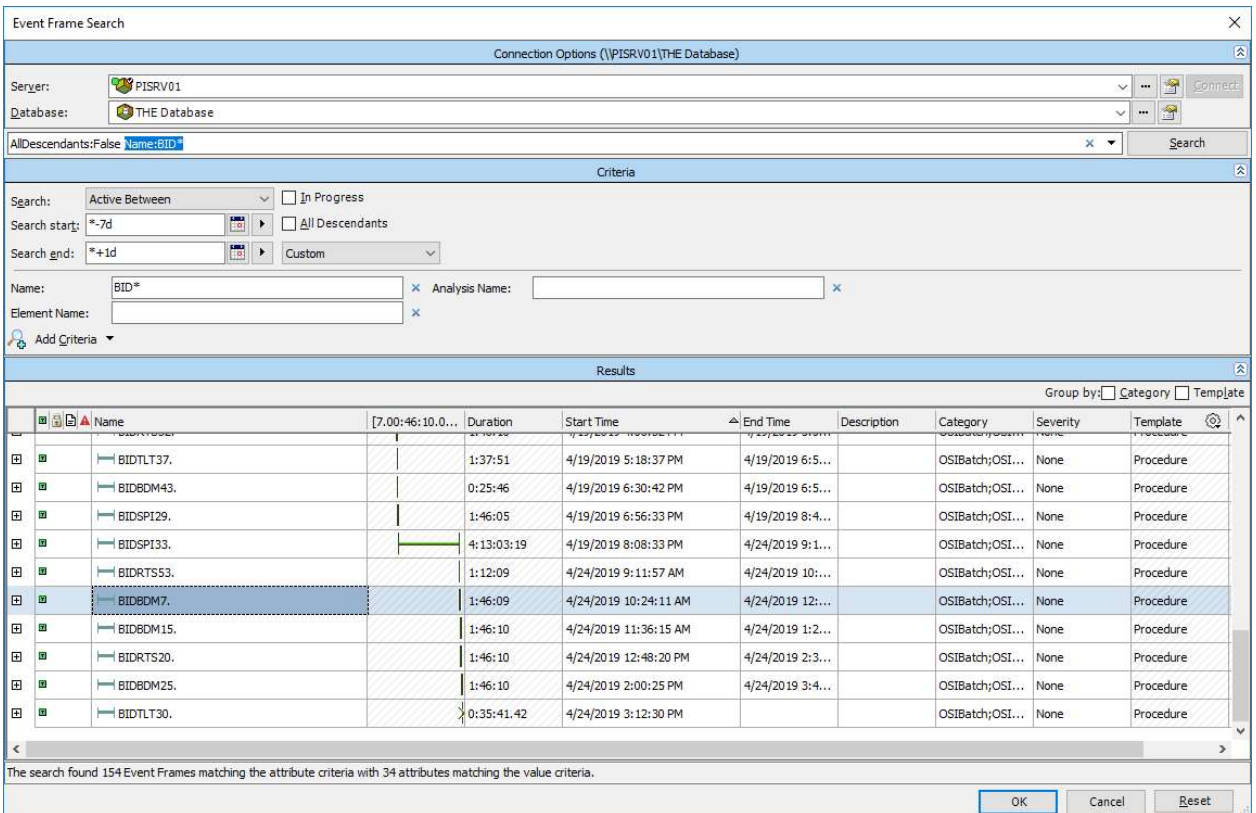
5. Select the CoverContent ► Insert ► Image ► Source: ..\Images\RtReports\_01.gif



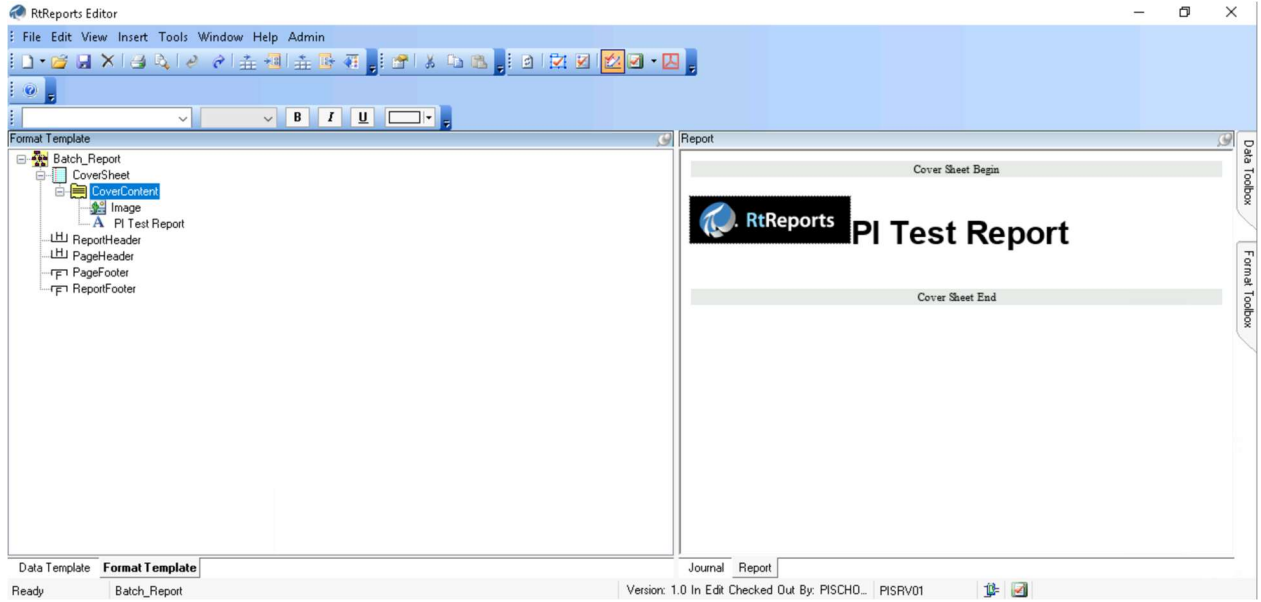
6. CoverContent Entry object ► Insert ► Text object ► Text: PI Test Report ► Font Section, Select Arial from Name dropdown menu, select 6 from the Size dropdown menu and select the "Bold" box ► OK



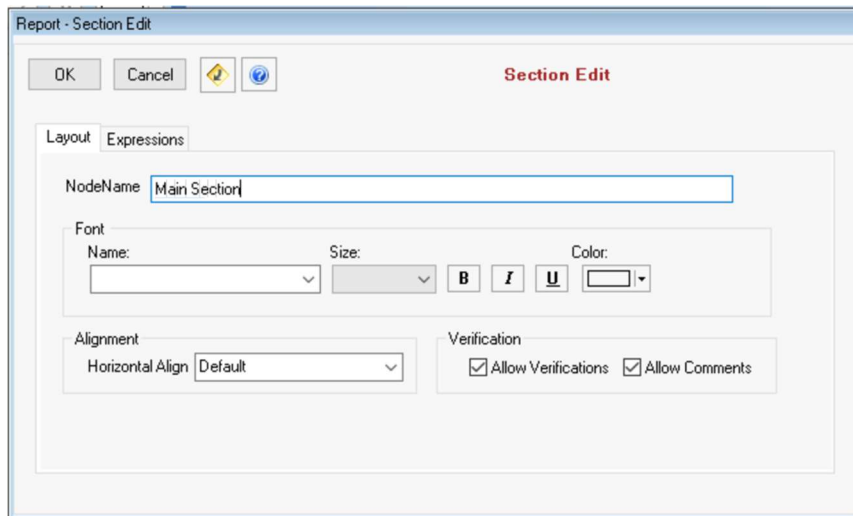
- To test the Batch\_Report: click View menu ► Test Report ► Use Event Frames or press F3 ► in the Event Frame Search prompt, expand Connection Options pane ► verify Server (AF Server) and Database; in the Criteria pane ► edit the Search criteria ► press Search ► in the Results pane, find and select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



- Switch to the Report tab to view the report display

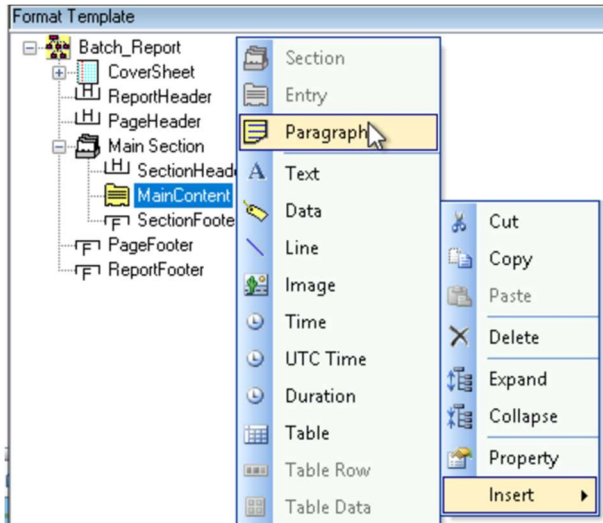


9. Back to the Format Template tab, select the root ▶ Insert ▶ Section ▶ Node Name: Main Section ▶ OK



10. Expand the Main Section ▶ double clicks Entry ▶ Node Name: MainContent
11. Select MainContent ▶ Insert ▶ Paragraph ▶ OK





12. Select Paragraph ► insert Text: Main Batch Data
13. MainContent ► *Insert* ► *Table* ► Node Name: BatchTable and set it to include 1 row and 3 columns ► switch to the *Border* tab ► *Frame: Border* ► Border Width: 1 ► *OK*
14. Expand the BatchTable, rename the *TableRow* to *Values*
15. Expand the Values row and rename each *TableData* respectively 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

Save Report Template and Refresh or F5

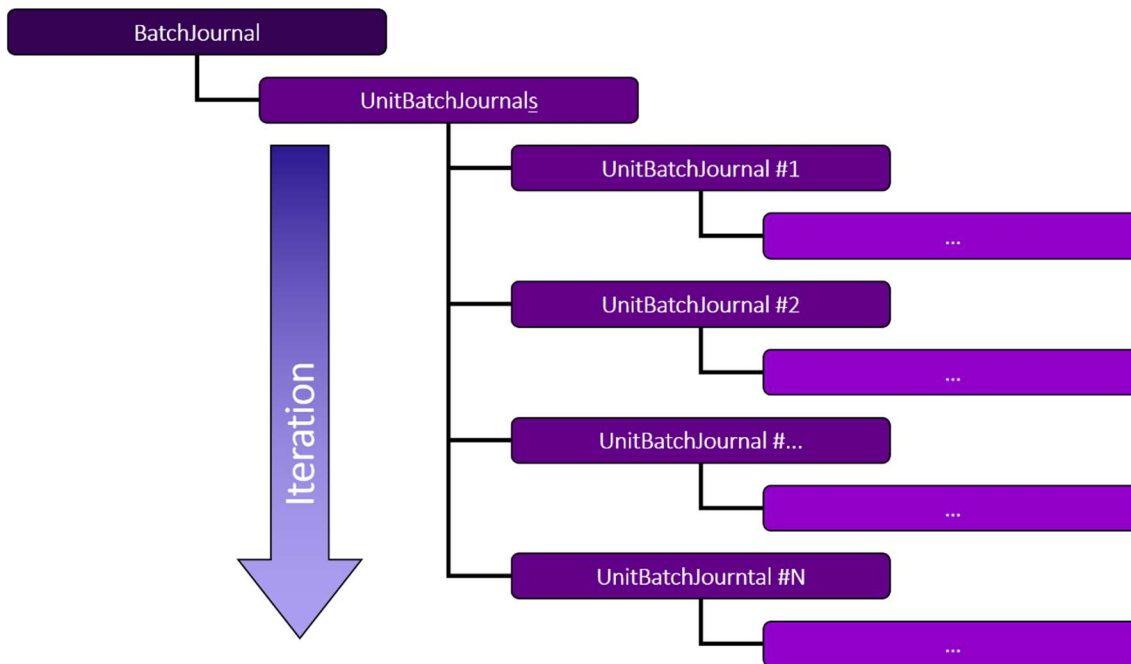
## 6. Using Iterations

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.

### 6.1 Background: 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.



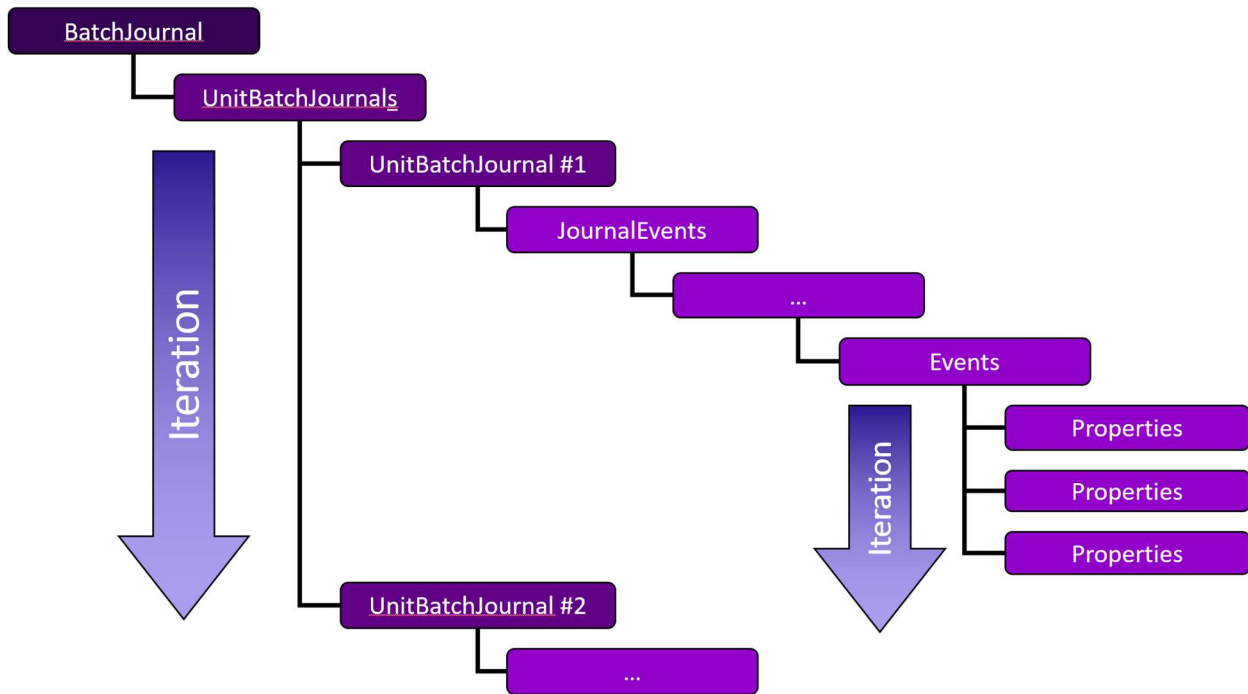
In this example, we are showing a BatchJournal with its own UnitBatchJournal. Typically, the UnitBatchJournals will hold one or many UnitBatchJournal, one for each PIUnitBatch under the PIBatch. Very often, the number of Journals included in a Journal is unknown. Nevertheless, we still need to process all the entries.

### 6.2 Background: Iterator Expression Entry

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.

### 6.3 Background: 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 UnitBatch running. This rule will yield to two sets of results, one for UnitBatch #1 and one for UnitBatch #2. If we want to display information for both UnitBatches, users will create a first iterator on the UnitBatchJournal, then further iterate on the Properties of the JournalEvents of each PUnitBatch.



Example with complete path to the data:

```
BatchJournal/UnitBatchJournals/UnitBatchJournal[Name="Unit1"]/JournalEvents/JournalEvent[Name="JAction"]/TerminalStates/TerminalState[StateName="Direct_Action"]/JournalEntries/JournalEntry/Results/Result[Label="CompressData"]/Value/TraceSet[@Name="CompressData.Tag"]/Events/Properties[position()=1]/Property[@Descriptor="22000"]
```

Iterator on UnitBatchJournal

```
BatchJournal/UnitBatchJournals/UnitBatchJournal
```

Iterator on the Events:

```
JournalEvents/JournalEvent[Name="JAction"]/TerminalStates/TerminalState[StateName="Direct_Action"]/JournalEntries/JournalEntry/Results/Result[Label="GetCompressData"]/Value/TraceSet[@Name="CompressData.Tag"]/Events/Properties
```

Actual Value:

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

## 6.4 Hands-on: Use Iterators



Before reading this section, please refer to the following course YouTube video:  
[RtReports Administration - Using iterations](#)

### 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

In the Format Template, insert an Entry “IterativeContent” under the “Main Section” section.

Insert a “UnitBatchTable” table under the IterativeContent entry

Identify that there are multiple UnitBatches.

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

**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*

## Preview

The screenshot displays the RtReports Editor interface. On the left, the 'Format Template' tree shows a hierarchical structure of report components. The 'Main Section' is expanded to show 'UnitBatchTable', which contains a 'Values' row with columns for 'UnitName', 'StartTime', and 'EndTime'. The 'UnitBatchTable' is further expanded to show 'Labels' with columns for 'UnitName', 'StartTime', and 'EndTime'. The 'Report' preview on the right shows the rendered output, including the RtReports logo, the title 'PI Test Report', and two tables: 'Main Batch Data' and 'Unit Batch Data'.

**Main Batch Data**

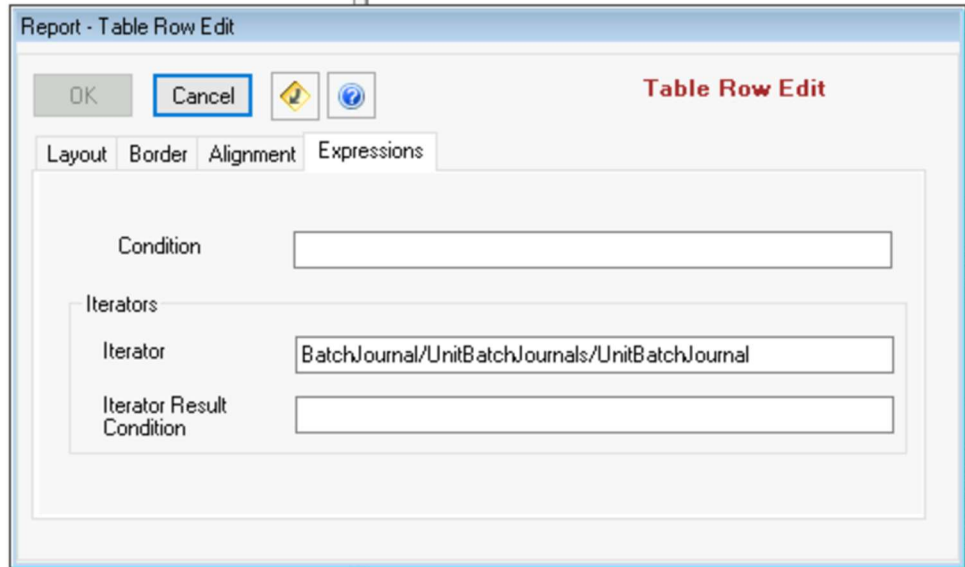
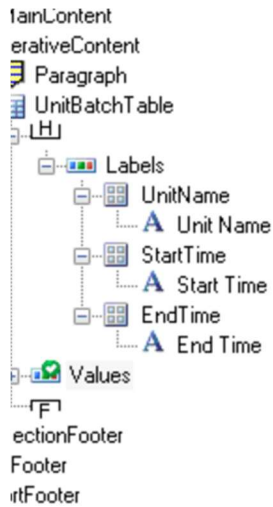
Batch ID	Start Time	End Time
BIDDEK2	29-May-22 10:59:02 AM	29-May-22 02:37:58 PM

**Unit Batch Data**

Unit Name	Start Time	End Time
R-1001	29-May-22 10:59:02 AM	29-May-22 12:05:07 PM
R-1002	29-May-22 11:39:07 AM	29-May-22 02:37:58 PM

### Step by step solution:

1. Login to RtReports Editor
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 ► Paragraph ► OK
5. Select Paragraph ► insert Text: Unit Batch Data
6. Select again IterativeContent ► Insert ► Table ► Node Name: UnitBatchTable ► Rows: 1 Columns: 3 ► switch to the Border tab ► Frame: Border ► Border Width: 1 ► OK.
7. Expand the UnitBatchTable ► rename the TableRow to *Values*
8. Expand the Values row and rename each TableData to UnitName, StartTime, and EndTime
9. Select the Values row ► copy ► select the table header ► paste ► rename the row to *Labels*
10. Expand the Labels row, select each cell ► Insert ► Text ► Text: same as the node name
11. 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



- Expand the Values row ► select the PIUnitName 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

The screenshot displays the AVEVA PI RtReports 4.1 Administration interface. On the left is a report tree with the following structure:

- port
- Sheet
- overContent
- thHeader
- header
- Section
- actionHeader
- mainContent
  - relativeContent
    - Paragraph
    - UnitBatchTable
      - UnitBatchJournal
        - UnitBatchJournal
          - SubBatchJournal
            - Empty>false</Empty>
            - BatchID>BIDDEK2.</BatchID>
            - Product>PrBDM2.</Product>
            - ProcedureName>RCDEK2.</ProcedureName>
            - PIUnitName>R-1001</PIUnitName>

- Footer

On the right, a 'Report - Data Edit' dialog box is open, titled 'Data Edit'. It contains the following fields and options:

- Buttons: OK, Cancel, Refresh, Help
- Layout section:
  - Node Name: Data
  - Numeric Format: (empty)
- Source: PIUnitName
- Font section:
  - Name: (empty)
  - Size: (empty)
  - Color: (empty)
  - Formatting: B, I, U, Color dropdown
- Expressions section:
  - Condition: (empty)

13. Save Report Template and Refresh or F5

## 7. 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.

Each Journal Action, there are two (2) methods for getting 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.

### 7.1 Background: Context

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

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

It is essential to select and insert Journal Actions at the correct/associated level, i.e., BatchJournal Action must be inserted in the Batch Level...

### 7.2 Background: Action Parameters and Results

#### 7.2.1 Action Parameters

An Action Parameter is a way of defining a data source in RtReports. It allows users to reference custom name(s) in the Result Definitions, instead of TRUE Data Source names.

Supported Action Parameter types:

Literal	The value column is considered a static numeric or string data type depending on the contents entered
PI Tag	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 the chosen Type of the Action Parameter, a dialog box is invoked.

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

### 7.2.2 Trend Result Definition

The Result Definition of Trend retrieves and outputs to the Journal Event the data and information of one or multiple tags for the Trend formatting object.

### 7.2.3 TraceSet Result Definition Types

Several result definitions invoke in the Data Journal a TraceSet, delivering their results with a tabular format, containing **Columns**, **Events**, and other **Properties**.

This part of the Journal provides all information automatically when used with the Format Template object **TraceSet** – however, it is important to note that auto-formatting provides less flexibility with table design; a **Table** may provide more useful formatting options, also when aggregating multiple results. Some TraceSet results are:

**Profile** – retrieves and outputs to the Journal Event the **Start Value**, **End Value**, **Maximum Value**, and/or **Minimum Value** of one or multiple tags between the Context Start and Context End.

**Summary Value** – retrieves and outputs to the Journal Event a selection of calculated values for one or multiple tag between the Context Start and Context End: **Minimum**, **Maximum**, **Average**, **Total**, **Average**, **Standard Deviation**, **Range**, **Mean**, and/or **Event Weighted**.

### 7.3 Hands-on: Configure a Journal Action



Before reading this section, please refer to the following course YouTube video:  
[RtReports Administration - Configure 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 *PI Point CDT158* of a reactor under the *Batch* level to the existing *Batch\_Report* report template.

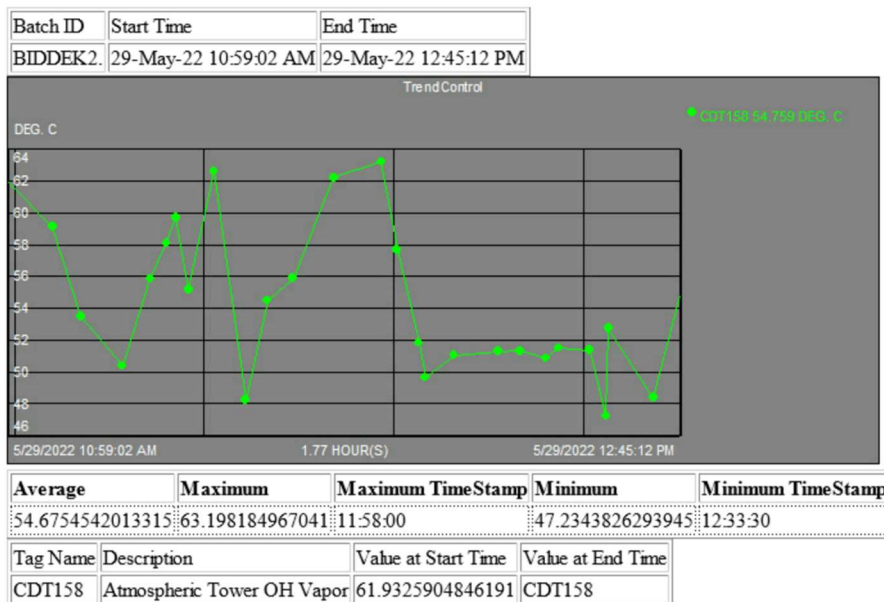
Display those *Result Definitions*.

#### Approach

- In the *Data Template*, 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

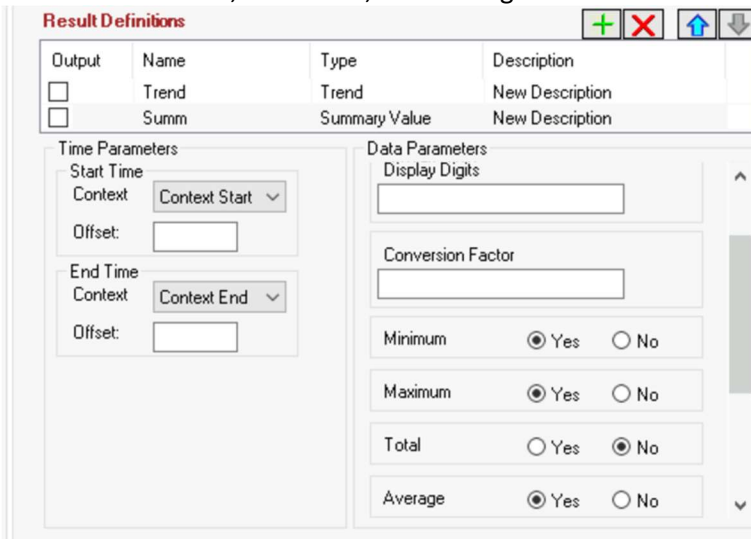
Main Batch Data



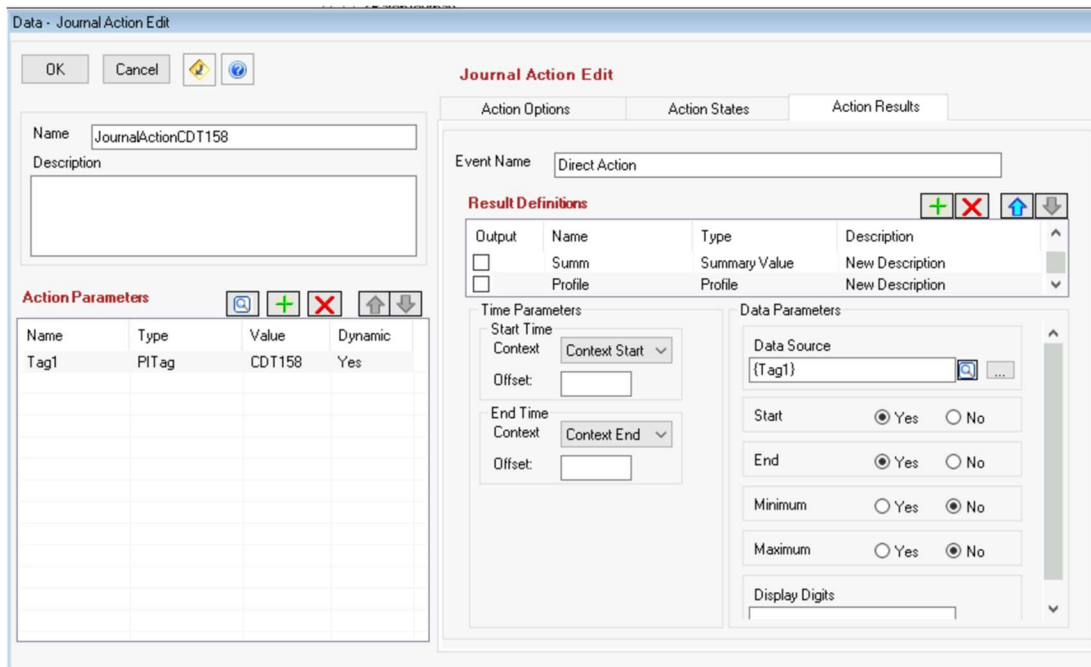
**Step by step solution:**

**Adding Journal Action to Data Template**

1. Login to RtReports Editor
2. Open the existing Batch\_Report template
3. In the Data Template tab, select Batch ► Insert ► Journal Action ► Name: "JournalActionCDT158"
4. Continue, under *Action Parameters* ► + ► Name: Tag1 ► Type: PITag ► browse tag CDT158 ► OK ► under *Journal Action Edit* ► *Action Results* tab...
5. ► + ► Name: Trend ► Type: Trend ► drag "Tag" (left) to Data source (right). Notice: { }
6. ► + ► Name: Summ ► Type: *Summary Value* ► drag "Tag" (left) to Data source (right) ► check "Yes" for Minimum, Maximum, and Average



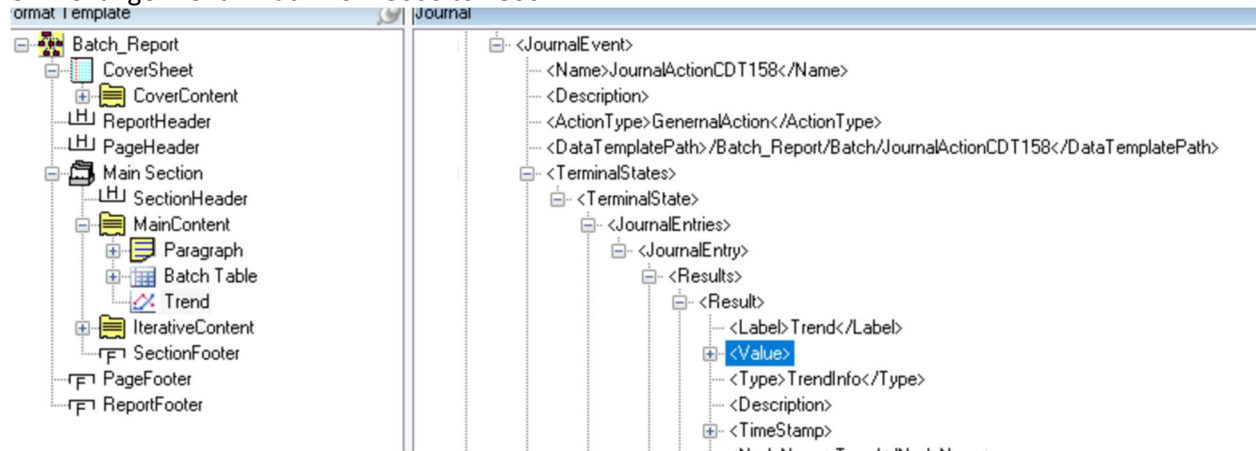
7. ► + ► Name: Profile ► Type: *Profile* ► drag "Tag" (left) to Data source (right) ► check "Yes" for Start and End ► OK



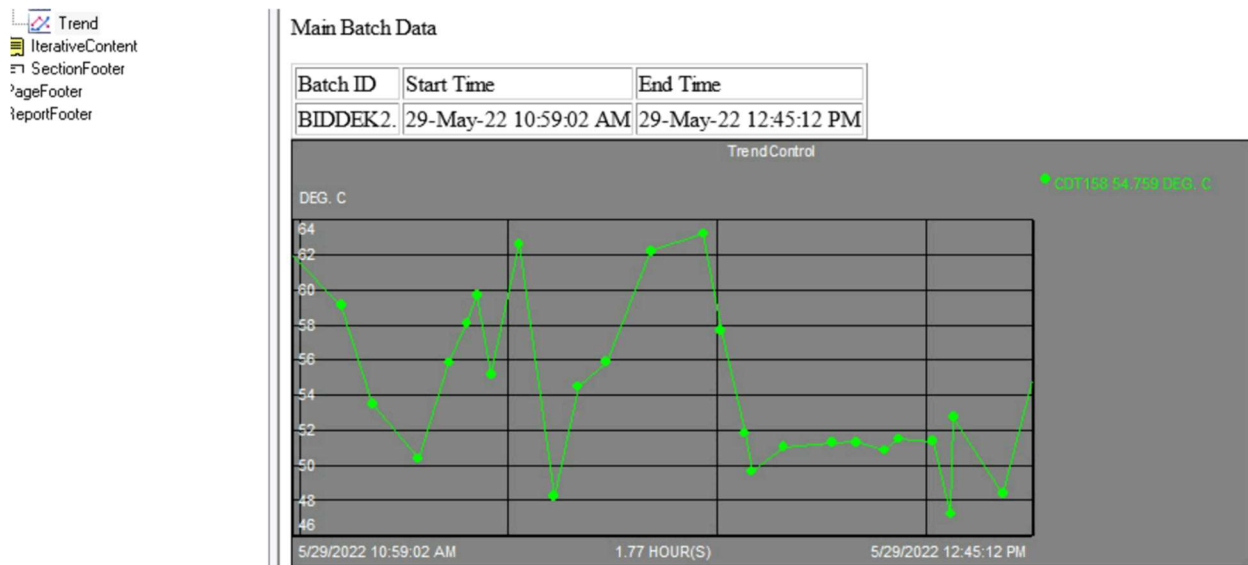
8. Save Report Template ► F3 to test or F5 to refresh

### Displaying the Trend

1. Switch to the Format Template tab ► expand the “Main Section”
2. Select MainContent ► Insert ► Trend object ► Node Name: Trend ► browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="JournalActionCDT158"]/TerminalStates/TerminalState[StateName="Direct\_Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Trend"]/Value* to the Source field
3. Change Trend Width from 8000 to 1800



4. Save Report Template and Refresh or F5

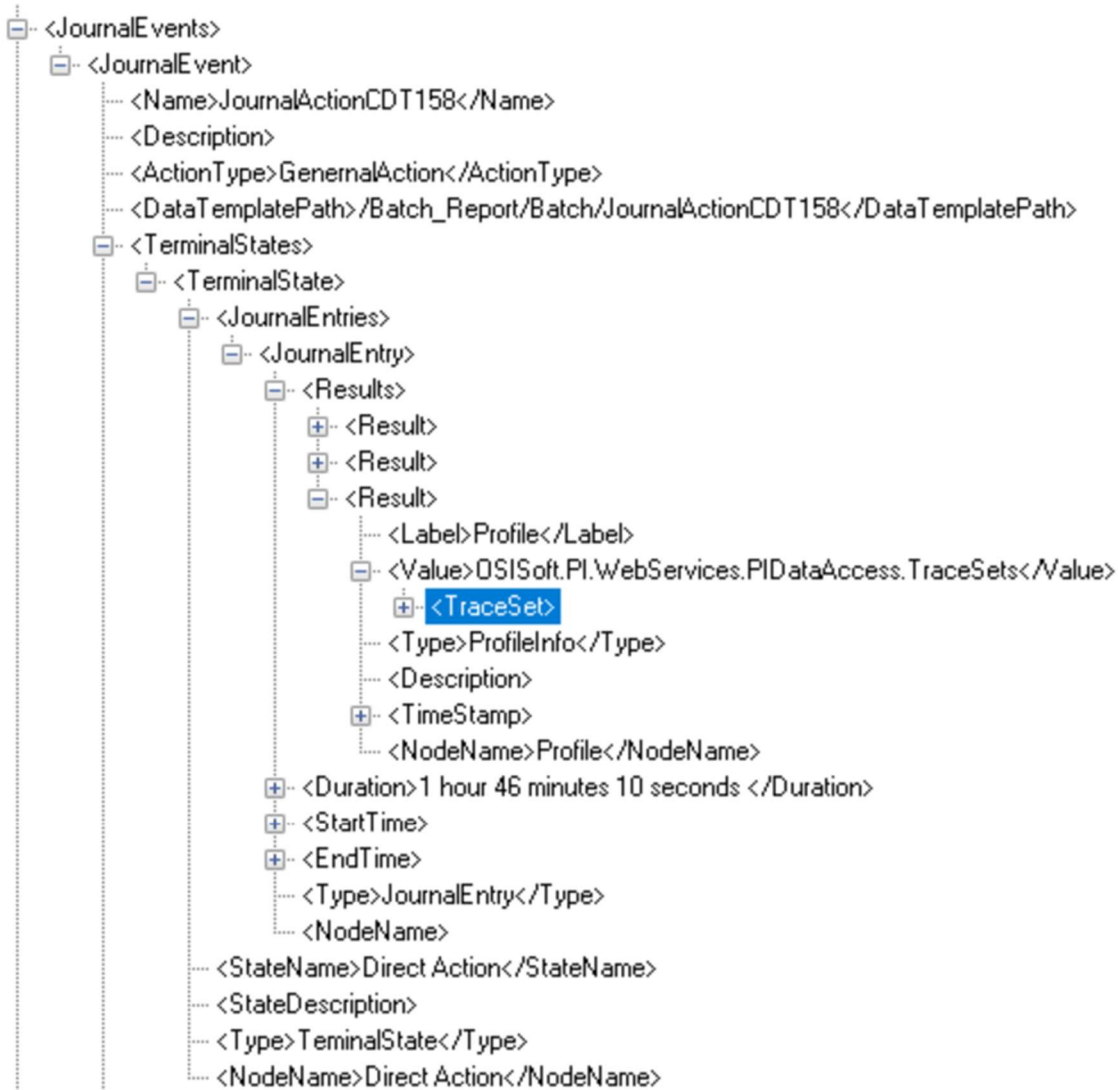


### Adding a TraceSet (smart table) for the Summary Data

1. In Format Template tab ► expand the “Main Section”
2. Select “MainContent” ► Insert ► Trace Set ► Node Name: Summary Values ► browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="JournalActionCDT158"]/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: Profile Table ► Rows: 1 ► Column: 4 ► Border tab ► Frame: Border ► Border Width: 1 ► OK
3. Expand the “Table - Profile Table ► rename TableRow to *Values*
4. Expand the *Values* row ► Column 1 – Column 4 ► Node Name: Tag Name, 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



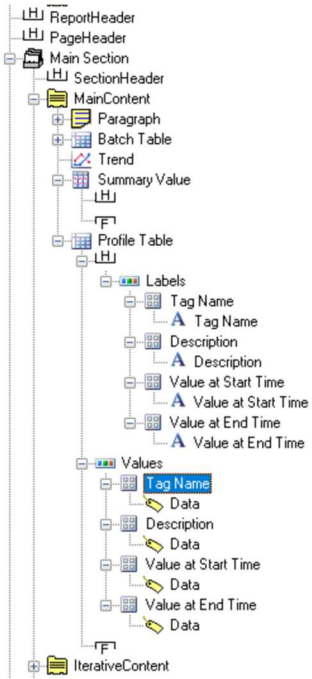
7. Expand the Value row ► select the *Tag Name* TableData ► Insert ► Data ► browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="JournalActionCDT158"]/TerminalStates/TerminalState[StateName="Direct\_Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Profile"]/Value/TraceSet[@Name="Profile.Summary"]/Properties/Property[@Descriptor="14"]* to Source field

8. Select the *Description* TableData ► Insert ► Data ► browse and drag *BatchJournal/JournalEvents/JournalEvent[Name="JournalActionCDT158"]/TerminalStates/Terminal State[StateName="Direct\_Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Profile"]/Value/TraceSet[@Name="Profile.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="JournalActionCDT158"]/TerminalStates/Terminal State[StateName="Direct\_Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Profile"]/Value/TraceSet[@Name="Profile.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="JournalActionCDT158"]/TerminalStates/Terminal State[StateName="Direct\_Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Profile"]/Value/TraceSet[@Name="Profile.Summary"]/Events/Properties[position()=1]/Property[@Descriptor="22009"]* to Source field

The screenshot displays a tree view of XML data. The following nodes are highlighted with red boxes:

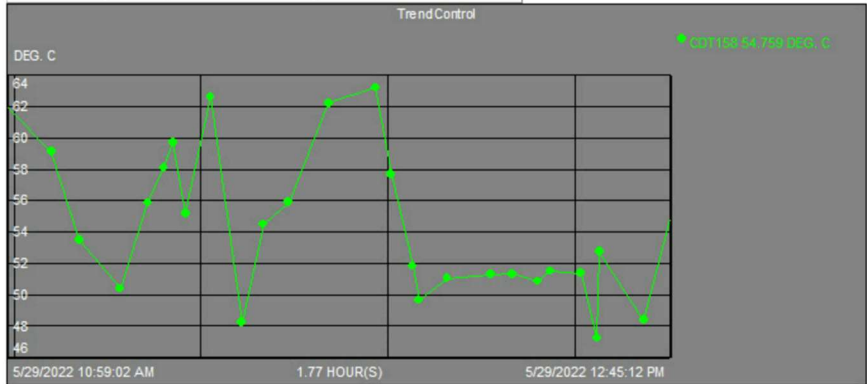
- Properties:**
  - <Property Descriptor=12>Atmospheric Tower OH Vapor</Property>
  - <Property Descriptor=14>CDT158</Property>
- Columns:**
  - <Name>End</Name>
    - <Description>End Value</Description>
    - <ColumnType>Float</ColumnType>
    - <PropertyColumn>22009</PropertyColumn>
    - <Category>Value</Category>
  - <Name>Start</Name>
    - <Description>Start Value</Description>
    - <ColumnType>Float</ColumnType>
    - <PropertyColumn>22008</PropertyColumn>
    - <Category>Value</Category>
- Events:**
  - <Property Descriptor=22008>6...9325904846191</Property>
  - <Property Descriptor=22009>5...7587699890137</Property>

11. Save the Report Template and Refresh or F5



Main Batch Data

Batch ID	Start Time	End Time
BIDDEK2.	29-May-22 10:59:02 AM	29-May-22 12:45:12 PM



Average	Maximum	Maximum TimeStamp	Minimum	Minimum TimeStamp
54.6754542013315	63.198184967041	11:58:00	47.2343826293945	12:33:30

Tag Name	Description	Value at Start Time	Value at End Time
CDT158	Atmospheric Tower OH Vapor	61.9325904846191	CDT158

## 7.4 Hands-on: Display Compressed Values



Before reading this section, please refer to the following course YouTube video:  
[RtReports Administration - Display Compressed Values](#)

### 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 AF Attribute* 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*

### Preview:

Format Template

- Batch\_Report
  - CoverSheet
  - CoverContent
  - ReportHeader
  - PageHeader
  - Main Section
    - SectionHeader
    - MainContent
    - IterativeContent
      - Paragraph
        - Unit Batch Data and Temperatures Measured
          - UnitBatchTable
          - CompValuesTable
            - Labels
              - Timestamp
              - Timestamp
              - Value
              - Value
            - Values
              - Timestamp
              - Time UTC
              - Value
              - Data
  - SectionFooter
  - PageFooter
  - ReportFooter

Report

Unit Batch Data and Temperatures Measured

Unit Name	Start Time	End Time
R-1001	29-May-22 10:59:02 AM	29-May-22 12:05:07 PM

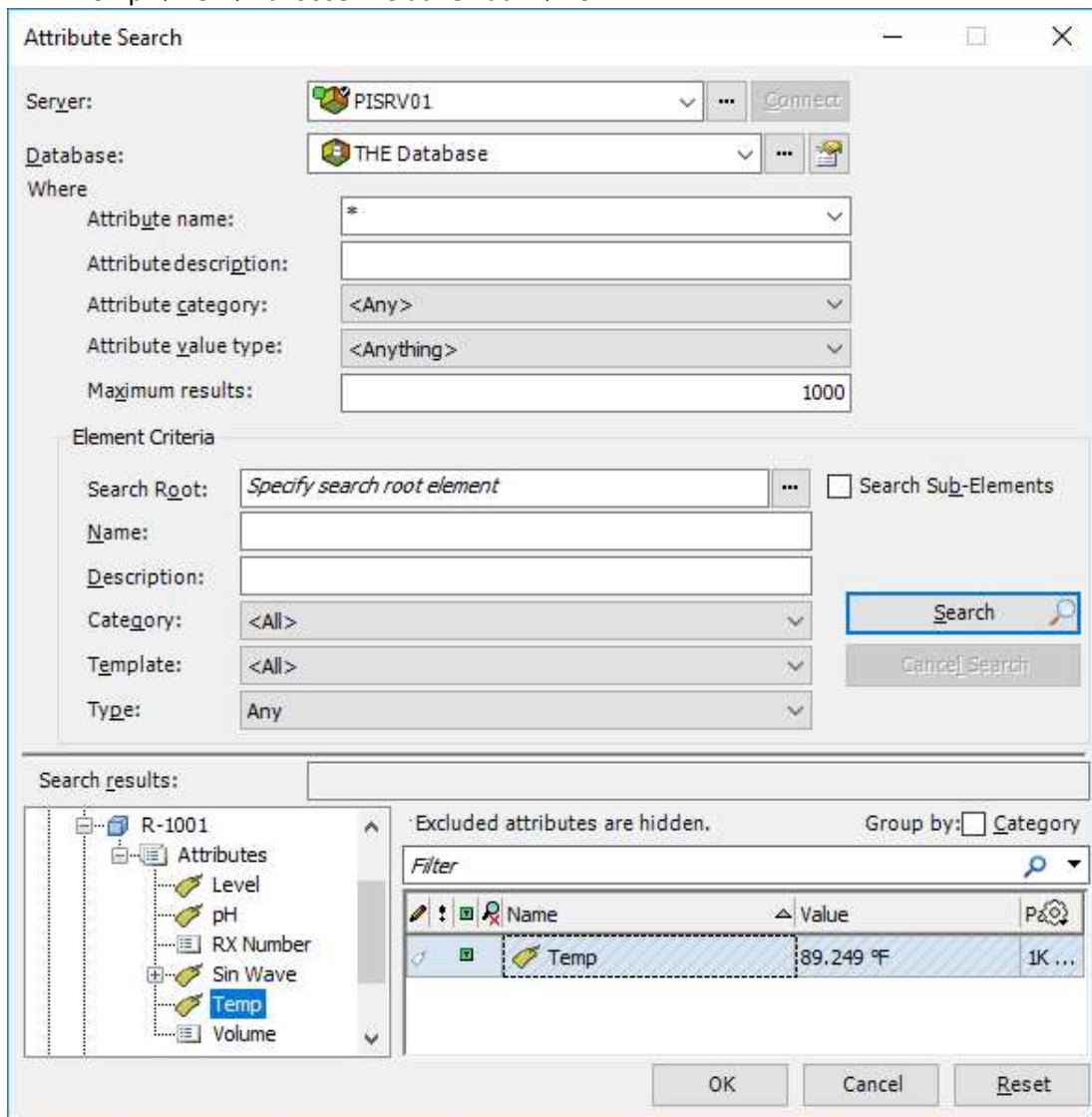
  

Timestamp	Value
29-May-22 10:59:56 AM	165.764541625977
29-May-22 11:01:26 AM	145.89924621582
29-May-22 11:01:56 AM	152.130661010742
29-May-22 11:05:26 AM	126.144538879395
29-May-22 11:12:56 AM	68.0779037475586
29-May-22 11:13:26 AM	50
29-May-22 11:15:26 AM	79.6814727783203
29-May-22 11:16:26 AM	75.7306137084961
29-May-22 11:22:26 AM	84.205207824707
29-May-22 11:25:26 AM	120.743553161621
29-May-22 11:33:56 AM	124.309669494629
29-May-22 11:39:57 AM	126.67626953125
29-May-22 11:40:56 AM	138.378463745117

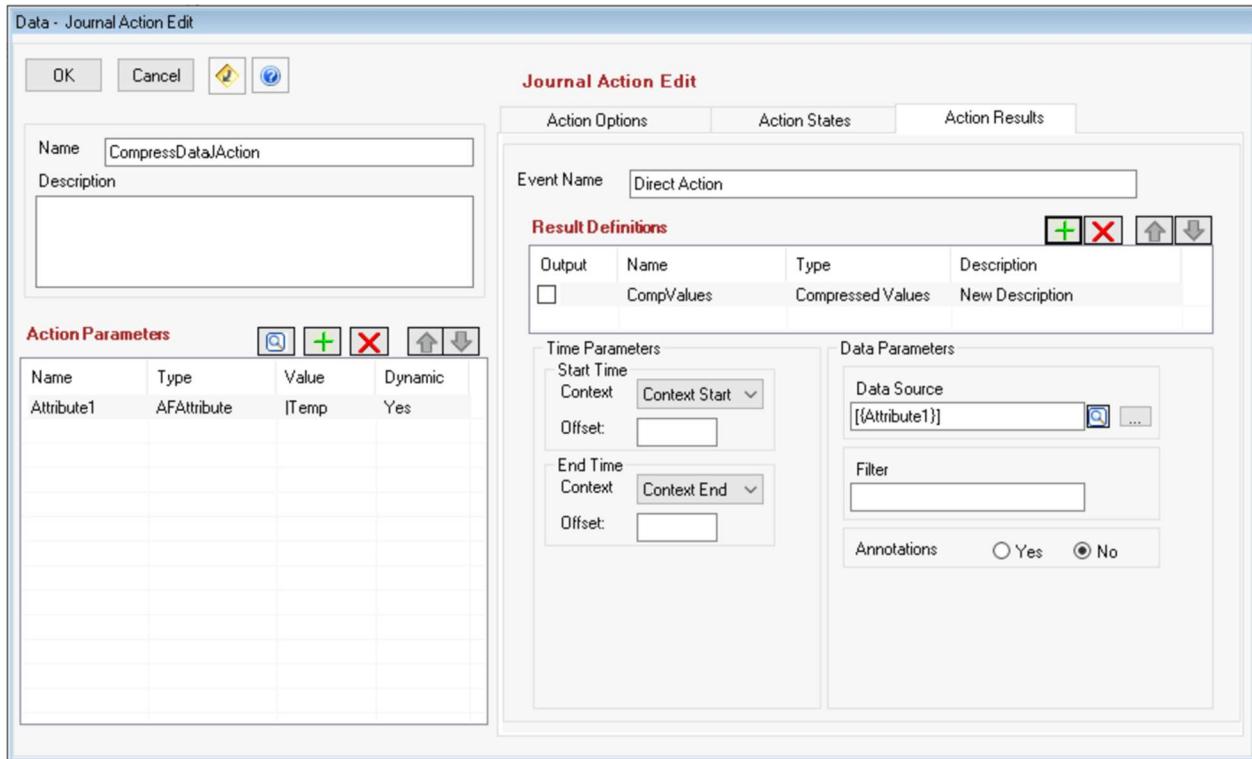


**Step by step solution:****Create a Journal Action for the extraction of the archived data**

1. Login to RtReports Editor
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: CompressDataJAction ► ...
6. Action Parameter ► + ► Name: Attribute1 ► 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 click IterativeContent ► Expressions tab ► paste *BatchJournal/UnitBatchJournals/UnitBatchJournal* into Iterator ► OK
3. (Optional) you can edit the text on the Paragraph to match our changes
4. Save Report ► Refresh or F5

Average	Maximum	Maximum TimeStamp	Minimum	Minimum TimeStamp
54.6754542013315	63.198184967041	11:58:00	47.2343826293945	12:33:30

Tag Name	Description	Value at Start Time	Value at End Time
CDT158	Atmospheric Tower OH Vapor	61.9325904846191	CDT158

Unit Batch Data

Unit Name	Start Time	End Time
R-1001	29-May-22 10:59:02 AM	29-May-22 12:05:07 PM

Unit Batch Data

Unit Name	Start Time	End Time
R-1002	29-May-22 11:39:07 AM	29-May-22 12:45:12 PM

### Adding a Table to display the compressed data

1. Select the Format Template tab
2. Expand Main Section
3. Select IterativeContent ► Insert ► Node Name: CompValuesTable ► 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

The screenshot displays the AVEVA PI Report Designer interface. On the left, the 'Format Template' tree shows the report structure, with 'CompValuesTable' expanded to 'Values'. On the right, the 'Journal' view shows the XML structure of the report. A 'Report - Table Row Edit' dialog box is open, showing the 'Expressions' tab. A red arrow points from the '<Properties>' node in the XML tree to the 'Iterator' field in the dialog box.

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

## 8. Building Time Reports

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

### 8.1 Background: Time Report Templates

To create a Time Report: Go to File ► New ► Time Report Template... ► and follow the Time Report Template Details dialog box.

Each user can define a fixed duration time template or a relative time template. A fixed duration time template creates a fixed 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.

#### 8.1.1 Recurrence

At the Recurrence Pattern section, the Recurrence drop-down list offers common time intervals recurrence pattern:

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

Once the Recurrence Pattern is selected, user will be prompted to further select desired report criteria(s).

***Official reports may be run only in the time range.***

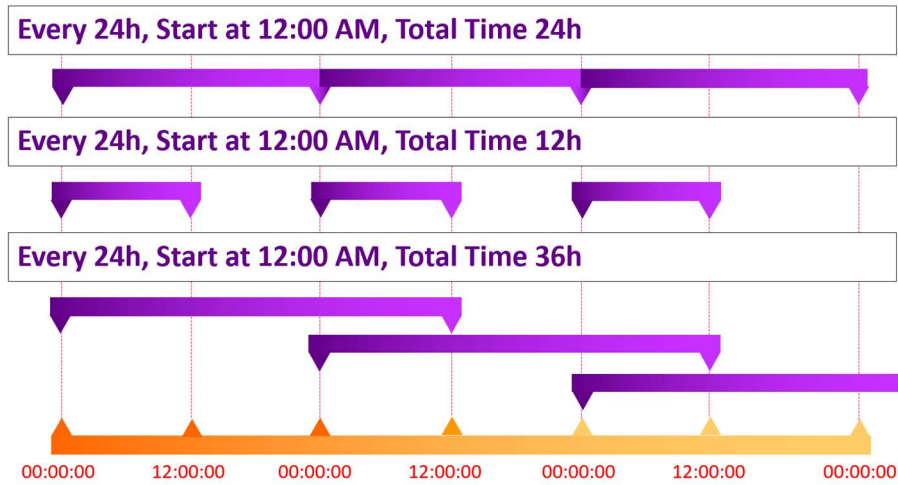
#### 8.1.2 Reference Start

A reference of time and/or date indicating when an official report begins. A calendar control will be prompted for user to select a date.

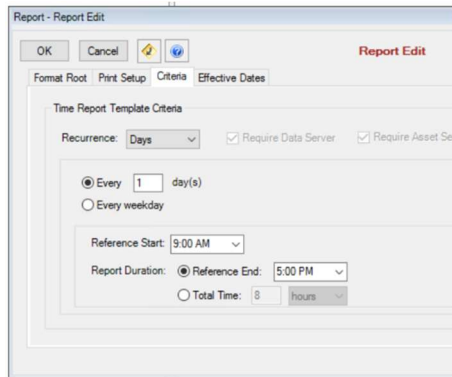
#### 8.1.3 Report Duration

For the duration of the report, it is possible to specify a Reference End time, or a Total Time can be specified for the right duration.

### 8.1.4 Applying Time Rules



### 8.1.5 Modify Time Rules



To change the Time Rules after creating the report:

- 1) Open the Time Report Template
- 2) Go to the Format Template pane
- 3) Right-click on root ► select Property
- 4) Switch to the Criteria tab

## 8.2 Background: Time Based Root Journal

A Time Report’s data 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 to extract and display the critical data in our reports.

At the top level, we find 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 can be found in each TimeBasedJournal.

## 8.3 Background: Using Equipment in a Time Report

Users can take advantage of an equipment context in a time-based report by using a Module Template or an Asset Context 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.

## 8.4 Hands-on: Configure a Time Report Template



Before reading this section, please refer to the following course YouTube video:  
[RtReports Administration - Time Based Reporting](#)

### Objectives

Discover Time Report Template

Discover Asset Context Template

### Problem Description

Configure a time-based report using AF Element Attribute information.

### Approach

Create a new time-based report.

Add an Asset Context Template RX under the Time Report Template.

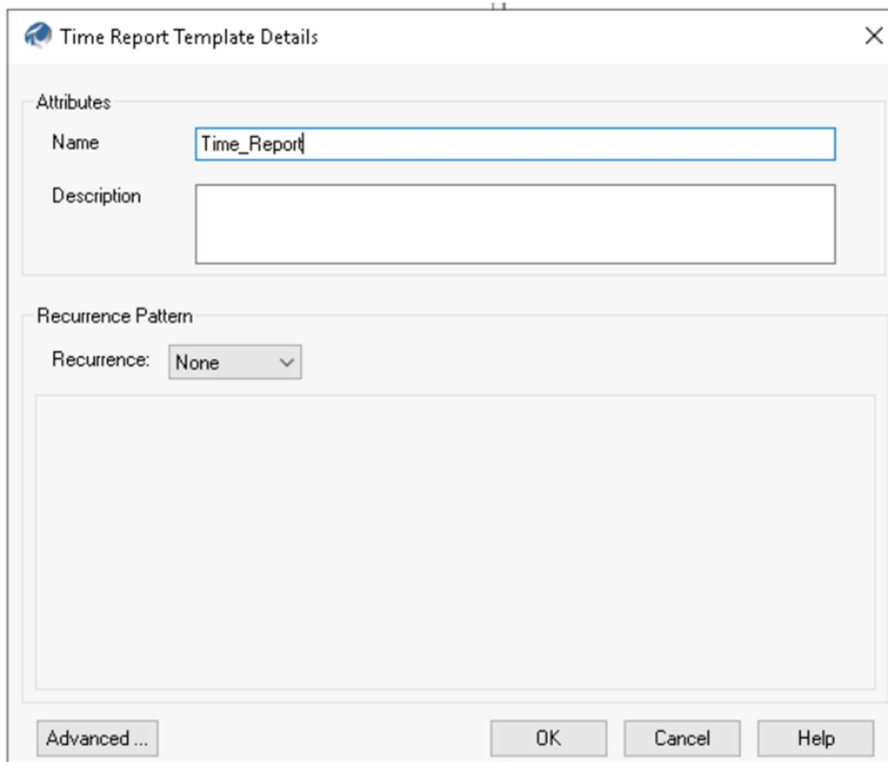
Add a Journal Action for Compressed Values Result Definitions of pH AF Attribute of the 1K Reactors.

Display the Unit Name, the AF Attribute and PI Point Names, and the Compressed Values.

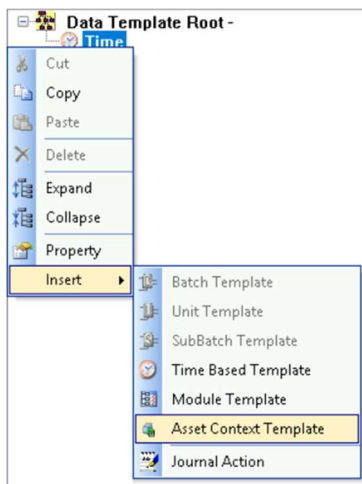
Attribute: pH	
Tag: \PISRV01\AT1002	
Value	Time
0.00	19:41:26
0.00	19:42:26
7.92	19:43:56
10.19	19:44:26
10.08	19:48:56
9.56	19:51:26
9.88	19:51:56

**Step by step solution:****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: *Time\_Report* ► Recurrence: *None* ► OK.

**Add an Asset Context Template RX under the Time Report Template.**

1. Switch to the Data Template tab ► select *Time* Template ► Insert ► Asset Context Template.



- Click on the magnifying glass next to the Template Name field. A search window should open up ► Select Server: *PISRV01*, Database: *THE Database*, Template: *RX* ► OK.

- Click on OK one more time.

### Add a Journal Action for Compressed Values Result Definitions of pH AF Attribute of the 1K Reactors.

- In the Data Template tab expand *Time* ► right-click on the *RX* template ► Insert ► Journal Action. Name: *CompValue*.
- Under **Action Parameters** click on the green plus icon “+” to add a new action parameter. For the Name type: *Attribute1*, select Type: *AFAttribute* ► Make sure you are connected to Server: *PISRV01* and Database: *THE Database*.
- In the Search Results expand 1K Reactors ► R-1001 ► Attributes ► select pH ► click on OK. Select Relative Path ► OK.



Attribute Search

Server: PISRV01

Database: THE Database

Where

Attribute name: \*

Attribute description:

Attribute category: <Any>

Attribute value type: <Anything>

Maximum results: 1000

Element Criteria

Search Root: Specify search root element   Search Sub-Elements

Name:

Description:

Category: <All>

Template: <All>

Type: Any

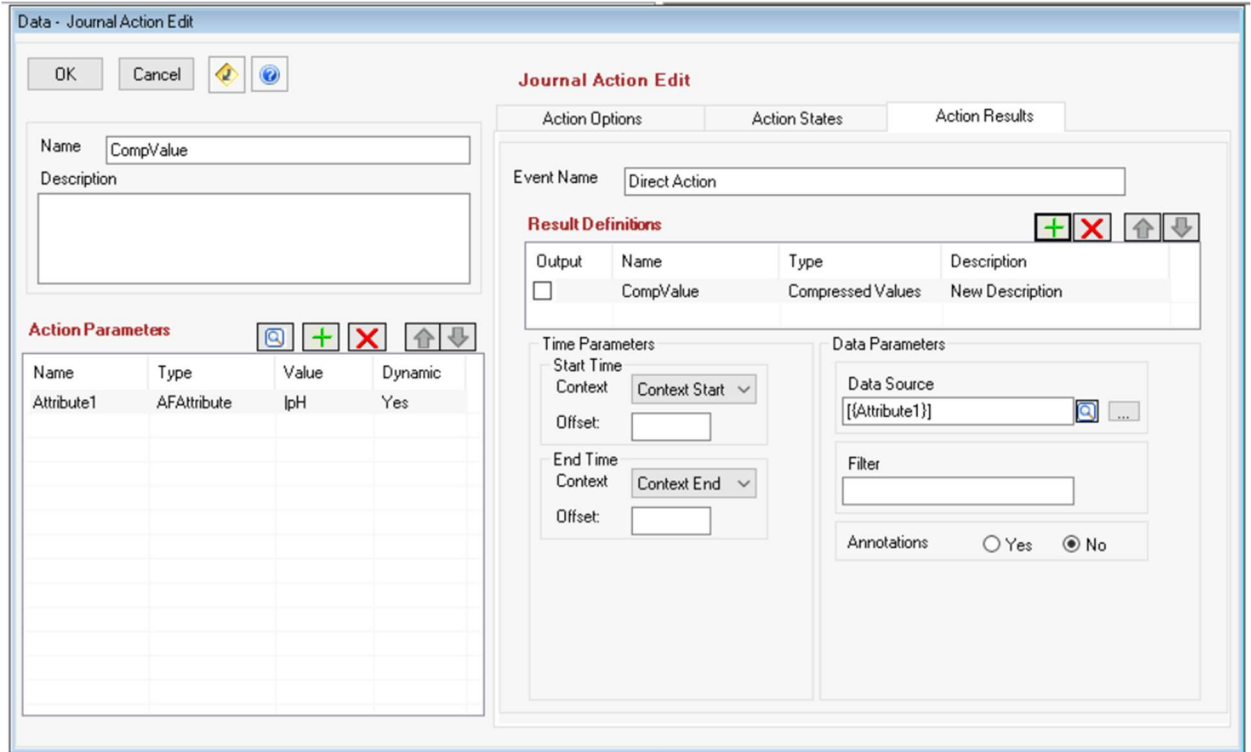
Search results:

Excluded attributes are hidden. Group by:  Category

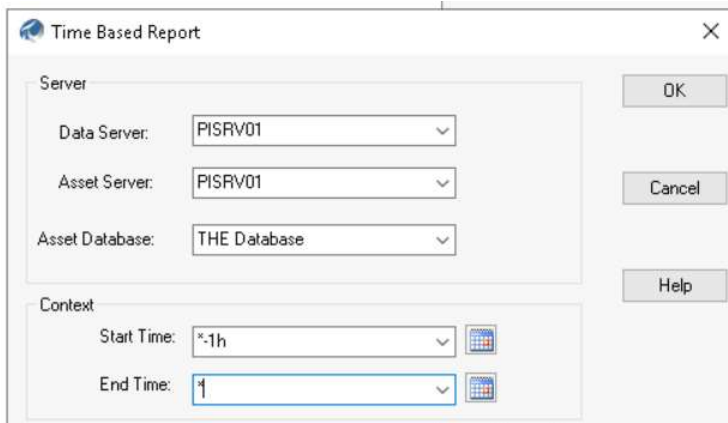
Filter

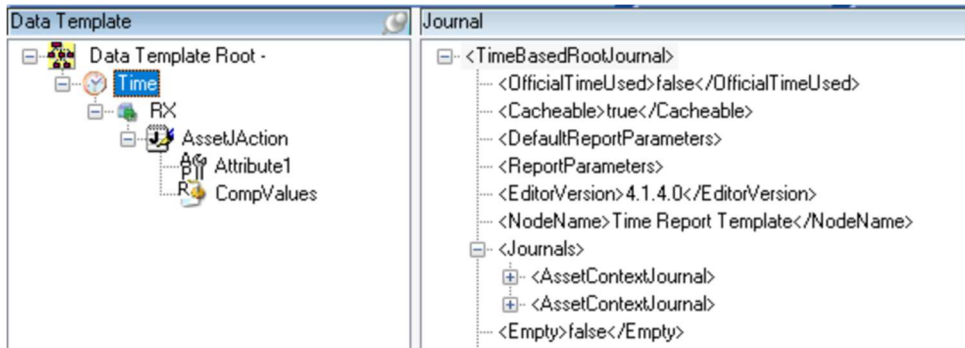
Name	Value	Path
pH	8.2273	1K Reactors R-1001 pH

- Go to Journal Action Edit ► Results Definitions tab and click on the green plus icon “+”. Name it: *CompValue*, Type: *Compressed Values*.
- Drag *Attribute1* (from the left) to the Data Source (on the right).



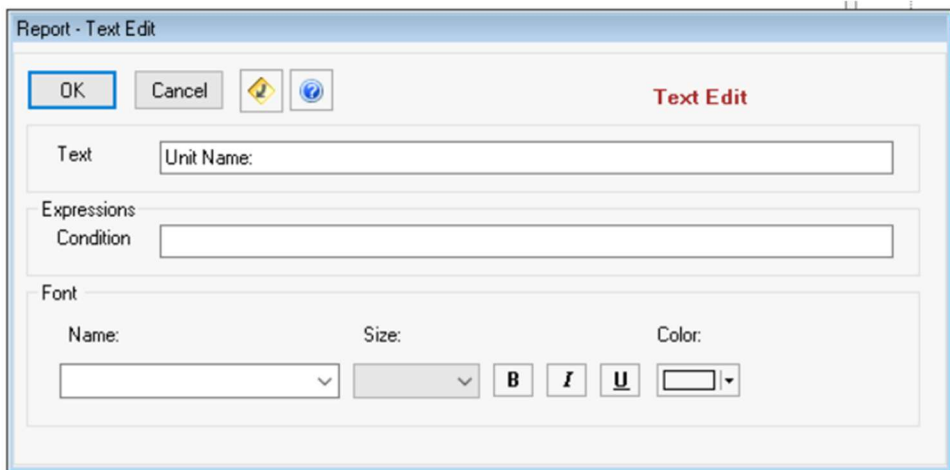
6. Click on OK.
7. Save and Test your Report ► View ► Test Report ► Data Server: PISRV01, Asset Server: PISRV01, Asset Database: THE Database, Start Time: \*-1h, End Time: \*.



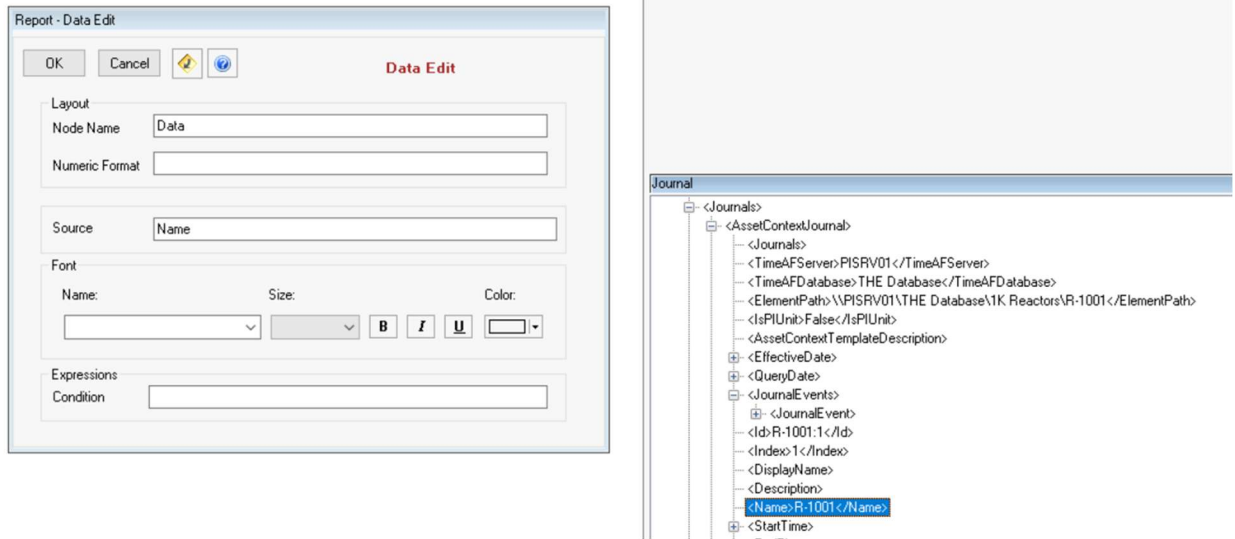


**Display the Unit Name, the AF Attribute and PI Point Names, and the Compressed Values.**

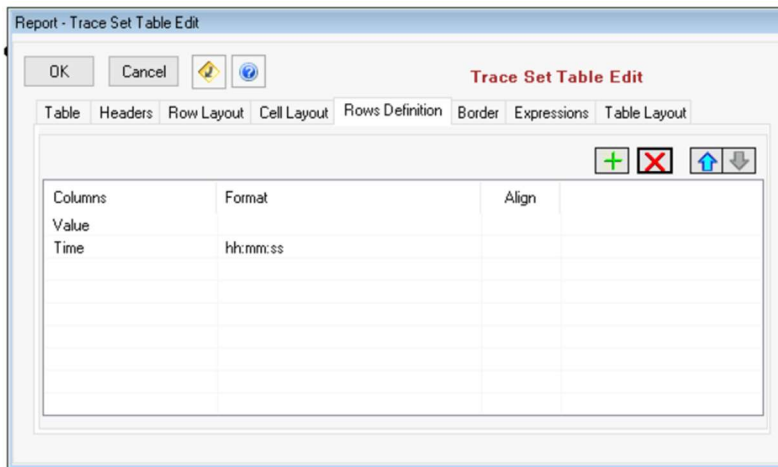
1. Go to Format Template tab ► Right-click on Time Report ► Insert ► Section. NodeName: RX Section.
2. Switch to the Expressions tab. Find and drag *TimeBasedRootJournal/Journals/AssetContextJournal* from the Journal to the Iterator field ► OK.
3. Expand RX Section ► right-click on Entry ► Insert ► Paragraph ► OK.
4. Right-click on Paragraph ► Insert ► Text: “Unit Name: ”



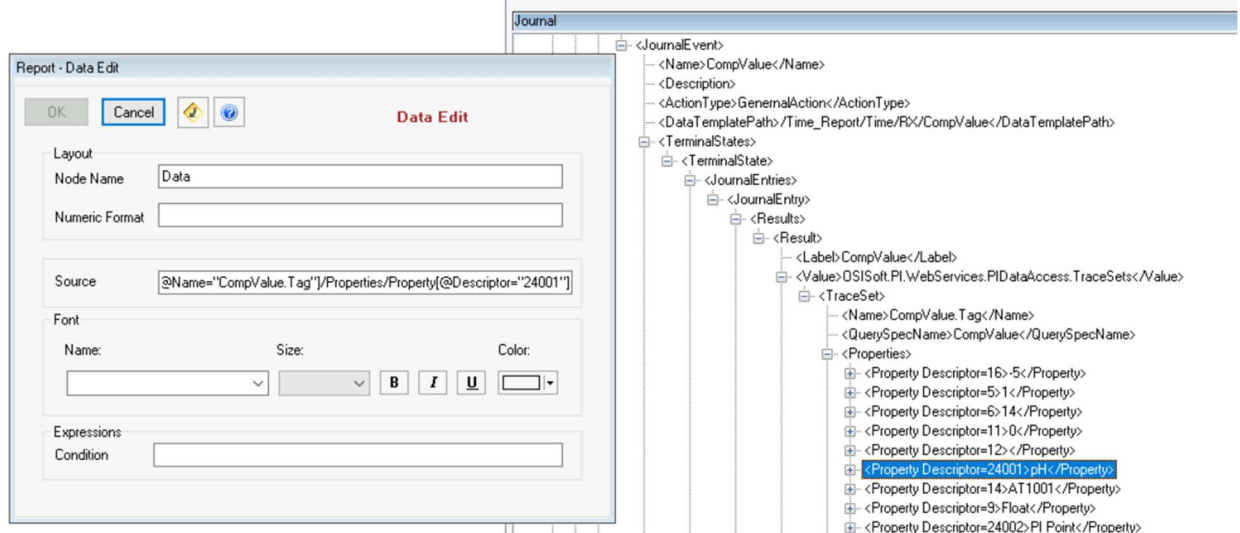
5. Right-click on Paragraph again ► Insert ► Data. Node Name: *Data*. Browse and drag the Unit name data item from the Journal to the Source field: *Name*.
6. Click on OK.



7. Right-click on Entry ► Insert ► Trace Set ► Node Name: pH Measurements Table
8. ► browse and drag to the Source field  
`JournalEvents/JournalEvent[Name="CompValue"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="CompValue"]/Value/TraceSet[@Name="CompValue.Tag"]`
9. ► Rows Definition tab ► red X to remove columns *ErrorCode*, *ErrorDescription*, and *IsGood*



10. ► Border tab ► Frame: Border ► Border Width: 1 ► OK
11. Right-click on the Table Header "H" under the *pH Measurements Table* ► Insert ► TableRow
12. ► right-click on TableRow ► Insert ► TableData
13. ► right-click on TableData ► Insert ► Text: "Attribute: "
14. ► right-click on TableData ► Insert ► Data ► find in the Journal and drag to the Source field:  
`JournalEvents/JournalEvent[Name="CompValue"]/TerminalStates/TerminalState[StateName="Direct Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="CompValue"]/Value/TraceSet[@Name="CompValue.Tag"]/Properties/Property[@Descriptor="24001"]` ► OK

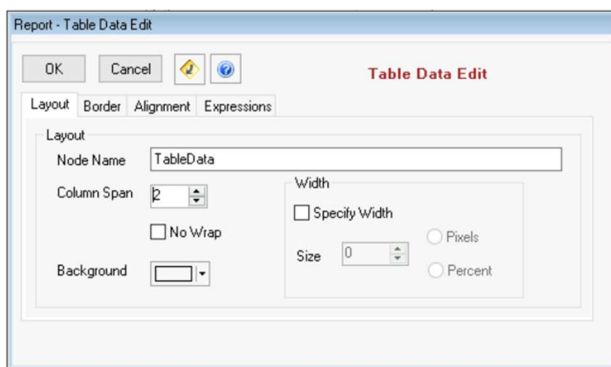


15. Right-click on the Table Header "H" under the *pH Measurements Table* ► Insert ► TableRow
16. ► right-click on TableRow ► Insert ► TableData
17. ► right-click on TableData ► Insert ► Text: "Tag: "
18. ► right-click on TableData ► Insert ► Data ► find in the Journal and drag to the Source field:  
 JournalEvents/JournalEvent[Name="CompValue"]/TerminalStates/TerminalState[StateName="Direct  
 Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="CompValue"]/Value/TraceSet[@Name="CompValue.Tag"]/Properties/Property[@Descriptor="24003"] ► OK
19. Save Report Template ► Refresh or F5

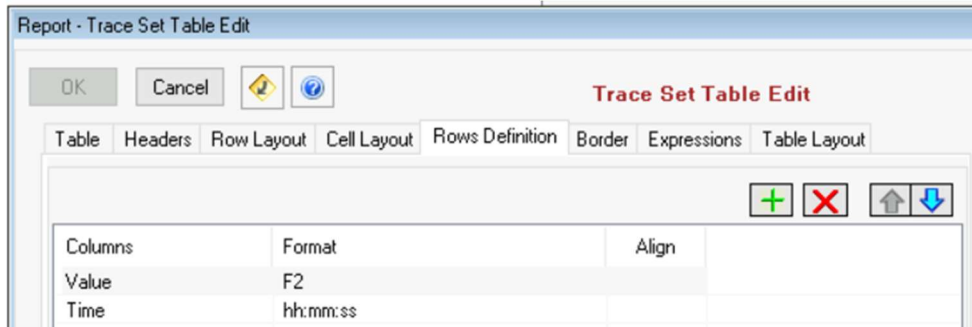
### Final modifications

Our report is ready, although you may notice that there are three key areas where improvements can be made. Firstly, let's address the spacing of the first two rows of the table, specifically the attribute and tag names. We will extend them so that they align with the rest of the table. Next, we'll rearrange the order in which the units are displayed to have them in ascending order, ensuring that unit R-1001 is displayed first. Additionally, we'll refine the decimal precision of pH values to two digits.

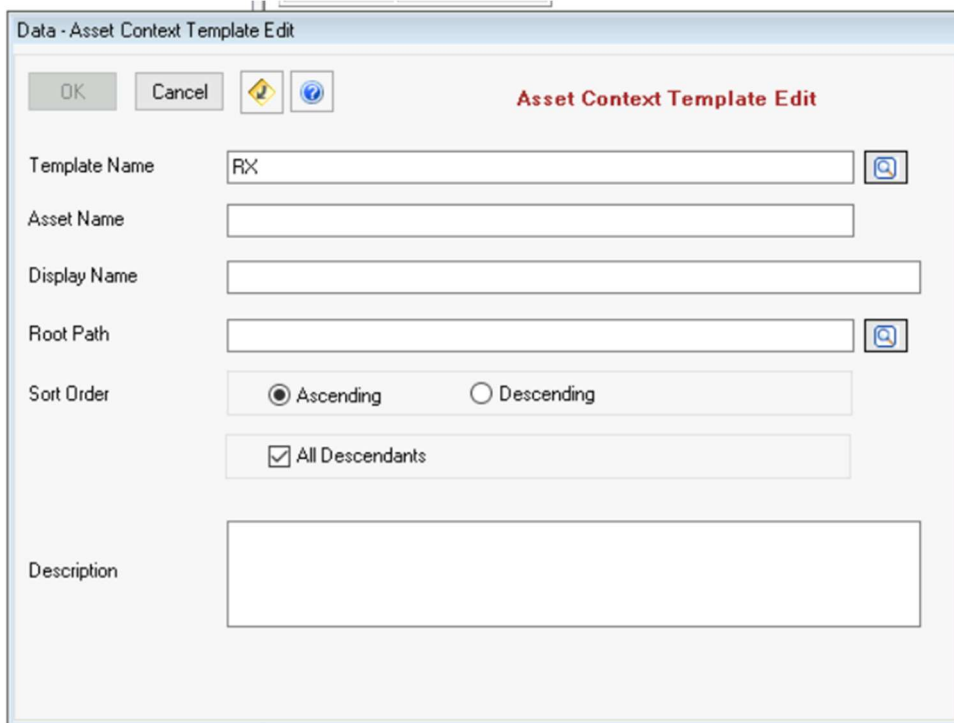
1. Switch to Format Template ► extend *RX Section* ► Entry ► *pH Measurements Table* ► TableRow ► Double-click TableData ► change the Column Span to 2.



2. Repeat the above steps for the second *TableData*.
3. In order to change the number of displayed decimal places, double-click the ph Measurements Table ► Rows Definition tab ► change the Value format to F2.



4. Switch to Data Template ► double-click the RX Template ► and select Ascending Sort Order.



5. Save your report and Refresh to see the changes.

Unit name: R-1002

Attribute: pH	
Tag: \PISRV01\AT1002	
Value	Time
0.00	19:41:26
0.00	19:42:26
7.92	19:43:56
10.19	19:44:26
10.08	19:48:56
9.56	19:51:26
9.88	19:51:56





**Step by step solution:****Create a new Time Report Template**

1. Login to RtReports Editor
2. Create a new template File ► New ► Time Report Template... ► Name: DailyShiftReport
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 ► Total Time: 1 days ► OK

**Time Report Template Details**

Attributes

Name: DailyShiftReport

Description:

Recurrence Pattern

Recurrence: Weeks

Recur every 1 week(s) on:

Sunday  Monday  Tuesday  Wednesday  
 Thursday  Friday  Saturday

Reference Start: 12:00 AM

Report Duration:  Reference End:   
 Total Time: 1 days

Advanced ... OK Cancel Help

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

1. Data Template tab ► select the "Time" Template ► Insert ► Time Based Template ► Name: 12h Intervals

**Data - Time Template Edit**

OK Cancel [Help Icon]

**Time Template Edit**

Layout: Time

Layout

Name: 12h Intervals

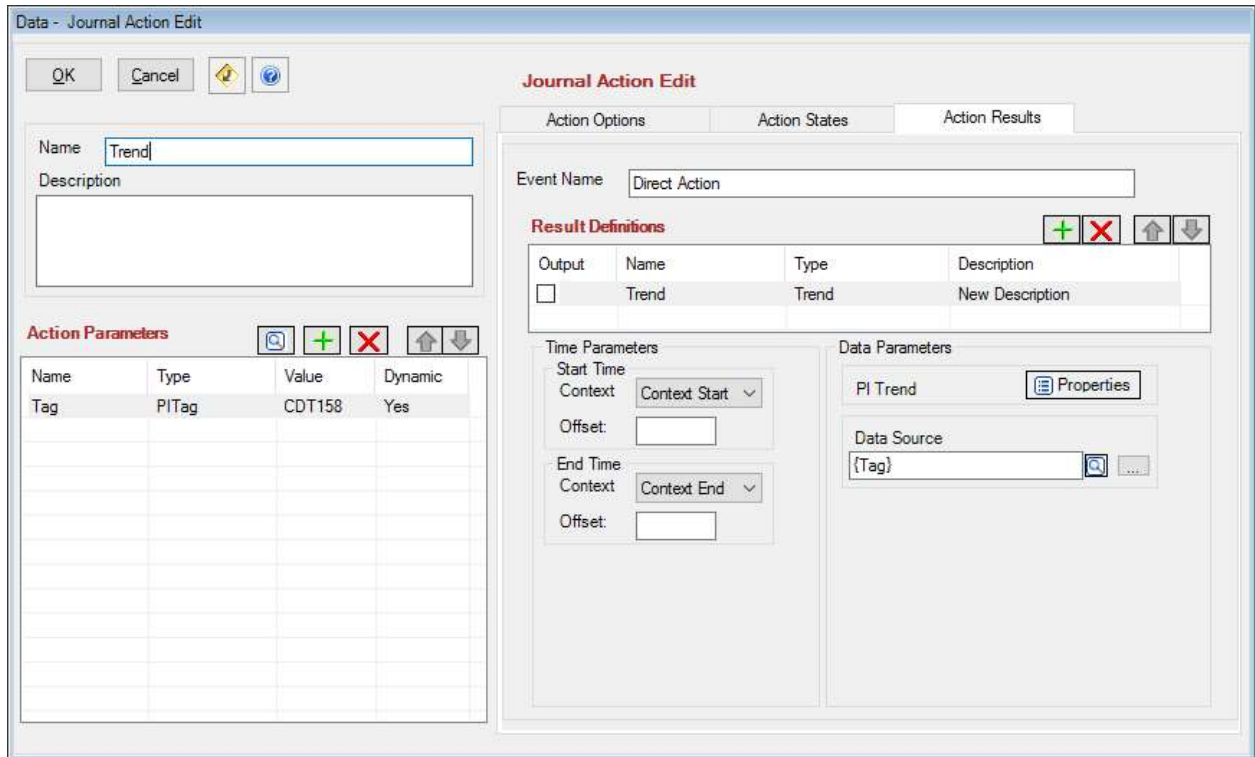
Display Name:

Description:

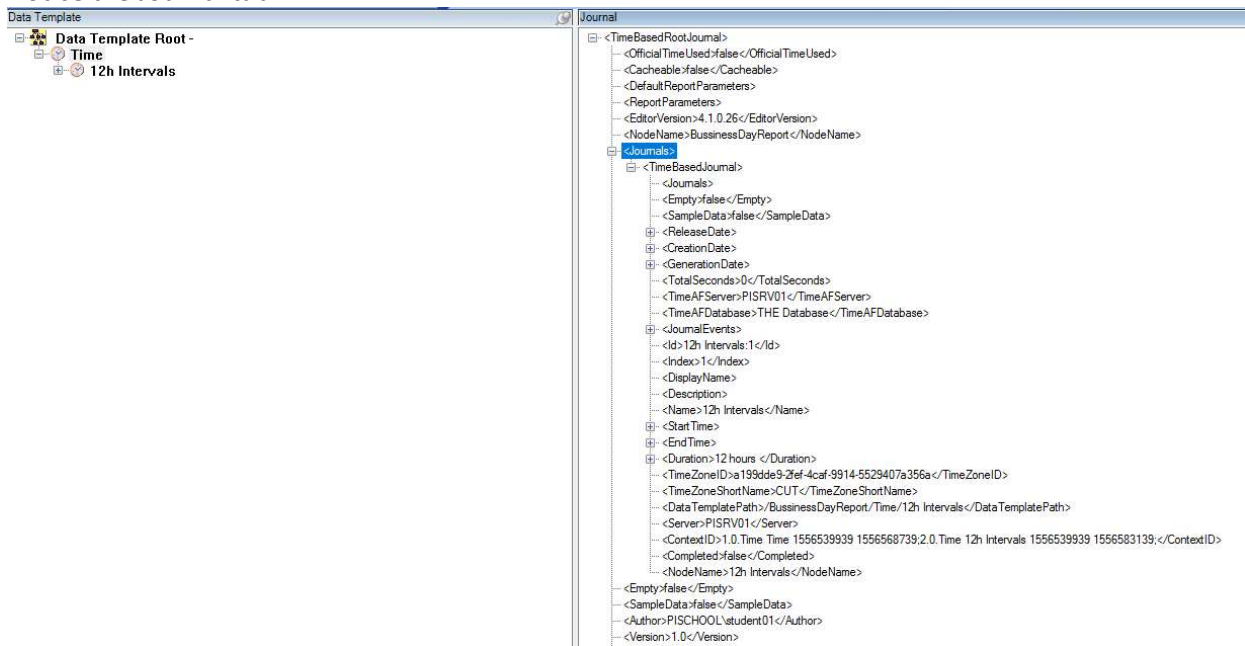
- Switch to the Time tab ► in the Sampling section ► Interval Number: 12 hours ► OK

The screenshot shows the 'Data - Time Template Edit' dialog box with the 'Time' tab selected. The 'Time Template Settings' section is expanded, showing 'Relative Time Span' selected. The 'Start Time Settings' section has 'Parent Start Time' selected for 'Start Time Context' and 'Offset' set to 0 days. The 'End Time Settings' section has 'Parent End Time' selected for 'End Time Context' and 'Offset' set to 0 days. The 'Sampling' section shows 'Interval Number' set to 12 and 'Interval Iterator' set to 'hour'.

- Expand "Time" and select the "12h Intervals" Time Based Template ► Insert ► Journal Action ► Name: Trend
- Action Parameter: + ► Name: Tag ► Type: PITag ► browse/find CDT158 > OK
- Journal Action Edit: Action Results tab
- Result Definitions: + ► Name: Trend ► Type: Trend ► drag "Tag" (left) to Data Source field (right) ► OK



7. Save ► Test the report, View ► Test Report... or F2 ► Data Server: PISRV01 ► Asset Server: PISRV01 ► Asset Database: THE Database ► Start Time: \*-8h ► End Time: \* ► OK
8. Notice the Journal tab



9. Switch to the Format Template tab ► select root ► Insert Section
10. Expand "Section" ► select "Entry" ► Property ► Expressions tab ► browse and drag *TimeBasedRootJournal/Journals/TimeBasedJournal* to Iterator field ► OK
11. Select Entry ► Insert ► Table ► Row: 2 ► Columns: 2 ► OK
12. Table Row 1 – Column 1 ► Insert ► Text: "Start Time: " ► OK
13. Table Row 2 – Column 1 ► Insert ► Text: "End Time: " ► OK

14. Table Row 1 – Column 2 ► Insert UTC Time ► browse and drag *StartTime/UTCSeconds* to Source field ► OK
15. Table Row 2 – Column 2 ► Insert UTC Time ► browse and drag *EndTime/UTCSeconds* to Source field ► OK
16. Select Entry ► Insert ► Trend ► browse and drag *JournalEvents/JournalEvent[Name="Trend"]/TerminalStates/TerminalState[StateName="Direct\_Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Trend"]/Value* to Source field ► OK

The screenshot displays the AVEVA PI RtReports 4.1 Administration interface. On the left, the 'Format Template' tree shows the structure of the report, including sections for 'Table Row 1' and 'Table Row 2'. The main area shows the 'Journal' data source tree, which is a hierarchical view of the data source. The 'Report - Trend Edit' dialog box is open, showing the 'Source' field with the path: *JournalEvents/JournalEvent[Name="Trend"]/TerminalStates/TerminalState[StateName="Direct\_Action"]/JournalEntries/JournalEntry[position()=1]/Results/Result[Label="Trend"]/Value*. The dialog also includes sections for 'Appearance' (Image, Height, Width, Margins) and 'Expressions' (Condition).

17. Save ► Refresh or F5

## 9. Condition Based Reporting

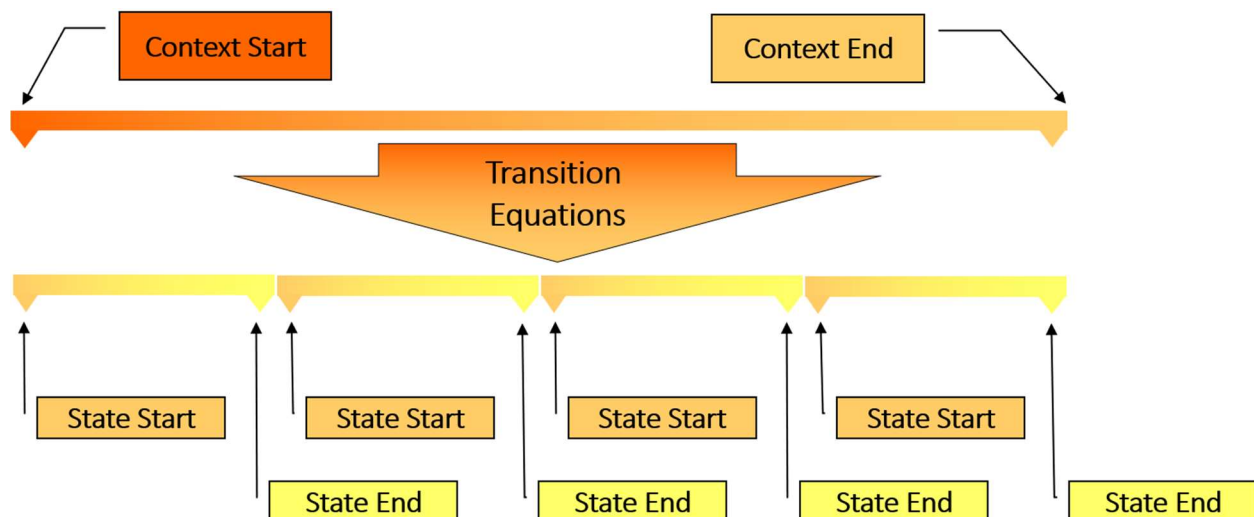
There are key advantages to setting up Conditions in Report Design. If a report is set to capture all information, it may capture a large volume of documents, with a significant risk of missing key information, overlooking small details and spending more time browsing unimportant data. With condition-based reports, we expect to save time, going right to the point and providing more details only when required.

The two (2) methods of Building Condition based report templates:

1. Configure **Action States** in the Data Template – Journal Action will extract only desired data. Example: extract and display values that are greater-equal to 55.5
2. Configure **Condition/Iterator Result Conditions** Criteria in the Format Template –all data will be extracted but display only what is desired. Example: display only values that are greater-equal to 55.5 (and hide all values that are less-than 55.5)

### 9.1 Background: Configure Action States (Data Template)

The context element defines the overall time range for the 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 Definition(s) for that state are generated.

Steps to configure Action States:

1. Insert a Journal Action.  
**Note:** A Journal Action may contain either Action Results or Action States (**never both**)
2. **Action States** tab: define total number of states. Click the "+" button to add a State and define the

- State Name
3. Define the Transition Equation in the form:
    - **IF {expression1} Then {statenumber1} Else If {expression2} Then {statenumber2} Else {statenumber3}**
    - Refer to PI Performance Equations for more information
  4. For each state, define Result Definitions
  5. **Action Options** tab: define the Interval of retrieval

**Journal Action Edit**

Action Options    Action States    Action Results

Name: TwoStatesJournalAction  
Description: If Active, print trend with Level Values  
If Inactive, output a message

Name	Type	Value	Dynamic
Source	PITag	BA:ACTIVE.1	Yes
Value	PITag	BA:LEVEL.1	Yes
InactiveMSG	Literal	Not Active	Yes

Index	State Name
1	Active
2	Inactive

Transition Equation: IF '{Source}'='Inactive' THEN 2 ELSE 1

Result Definitions:

Output	Name	Type	Description
<input type="checkbox"/>	Trend	Trend	New Description

Time Parameters:

Start Time Context: State Start  
Offset:   
End Time Context: State End  
Offset:

Data Parameters:

PI Trend Properties  
Data Source: {Value}

### 9.1.1 Timing of the Action State Transitions Evaluation

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** toggle will generate an error if the result cannot be computed.

Name	Type	Value	Dynamic
Source	PITag	BA:ACTIVE.1	Yes
Value	PITag	BA:LEVEL.1	Yes
InactiveMSG	Literal	Not Active	Yes
Interval	Literal	60	Yes

### 9.1.2 Filter or Suppress Transitions Evaluations

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.

### 9.1.3 More Action Options

Users may define Time Offset (dead band) from the Start Time and/or End Time (in seconds).

The Output Type 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.

## 9.2 Hands-on: Report by Exception with Action States



Before reading this section, please refer to the following course YouTube video:  
[RtReports Administration - Report by Exception with Action States](#)

### Objectives

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

### Problem Description

Monitor PI Tag “Ba:Level.1” and show only trend data when tank level is High.

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

Configure a new Batch Report Template “Exception\_Report”

Create a Journal Action using the Action States (State Machine) to configure the custom 3-States as defined in the following table. Evaluate the Transition Equation every 30 seconds.

Create a table to show the results.



**Report Preview:**

State	Start Time	End Time	Trend
High Alarm	29-May-22 10:59:02 AM	29-May-22 11:33:02 AM	
In Range	29-May-22 11:33:02 AM	29-May-22 11:36:32 AM	---
Low Alarm	29-May-22 11:36:32 AM	29-May-22 12:02:32 PM	---
In Range	29-May-22 12:02:32 PM	29-May-22 12:22:02 PM	---
High Alarm	29-May-22 12:22:02 PM	29-May-22 12:45:12 PM	

**Step by step solution:**

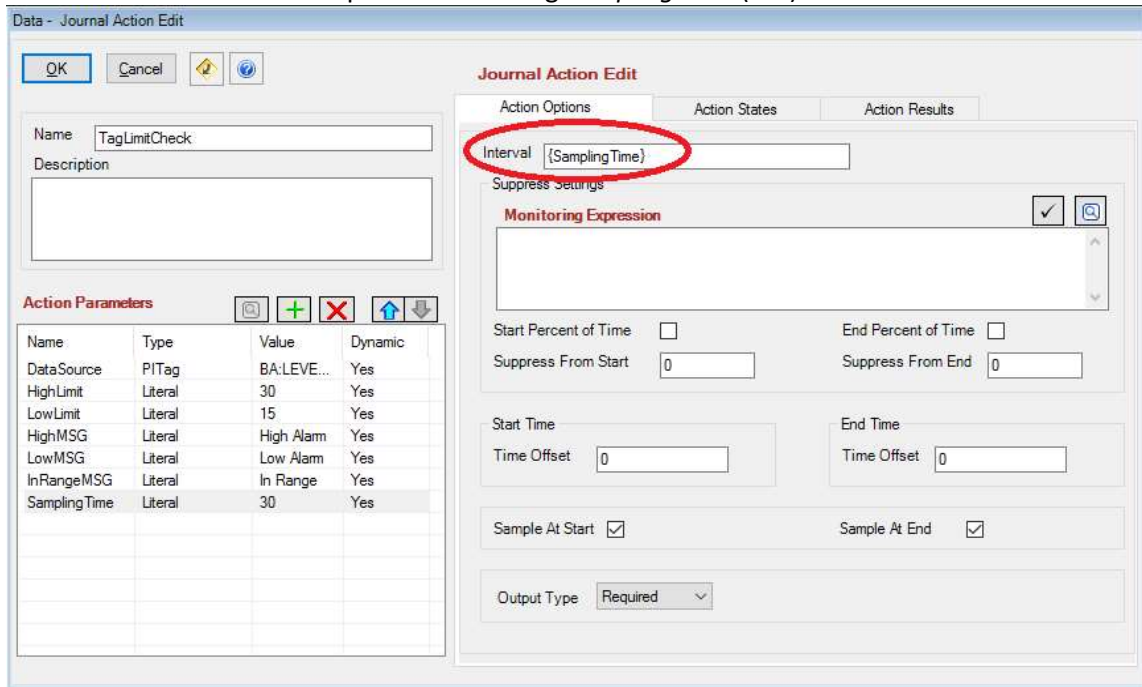
**Create the Action State**

1. Login to RtReports Editor
2. Create a new *Exception\_Report* template
3. In the Data Template, select the "Batch" ► Insert ► Journal Action ► Name: *TagLimitCheck*.
4. Action Parameters: + to add the following Action Parameters

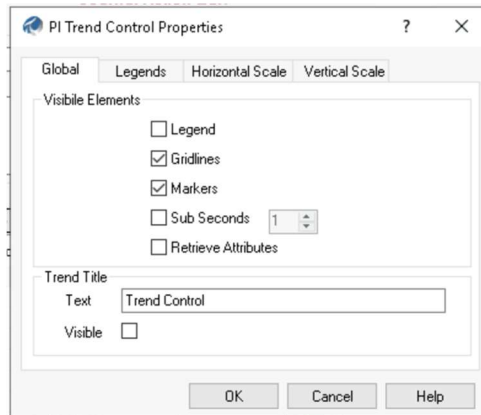
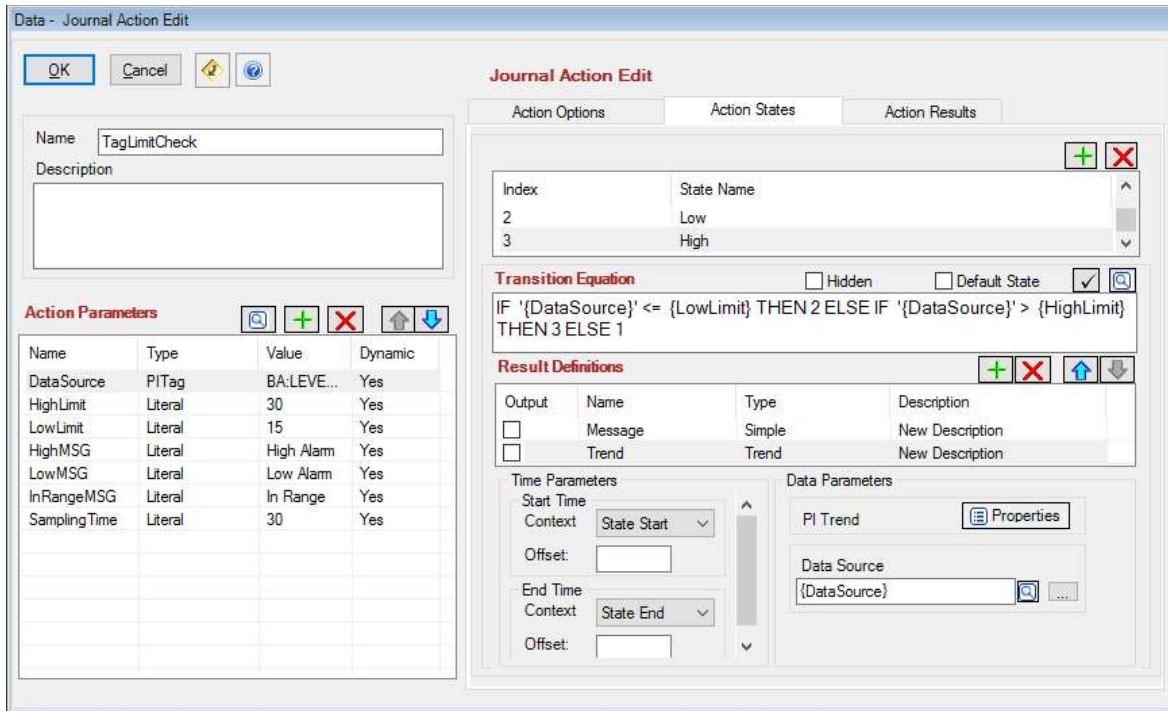
Name	Type	Value
DataSource	PI Tag	BA:Level.1
HighLimit	Literal	30
LowLimit	Literal	15

HighMSG	Literal	High Alarm
LowMSG	Literal	Low Alarm
InRangeMSG	Literal	In Range
SamplingTime	Literal	30

1. Journal Action Edit: Action Options tab ► drag *SamplingTime* (left) to Interval field



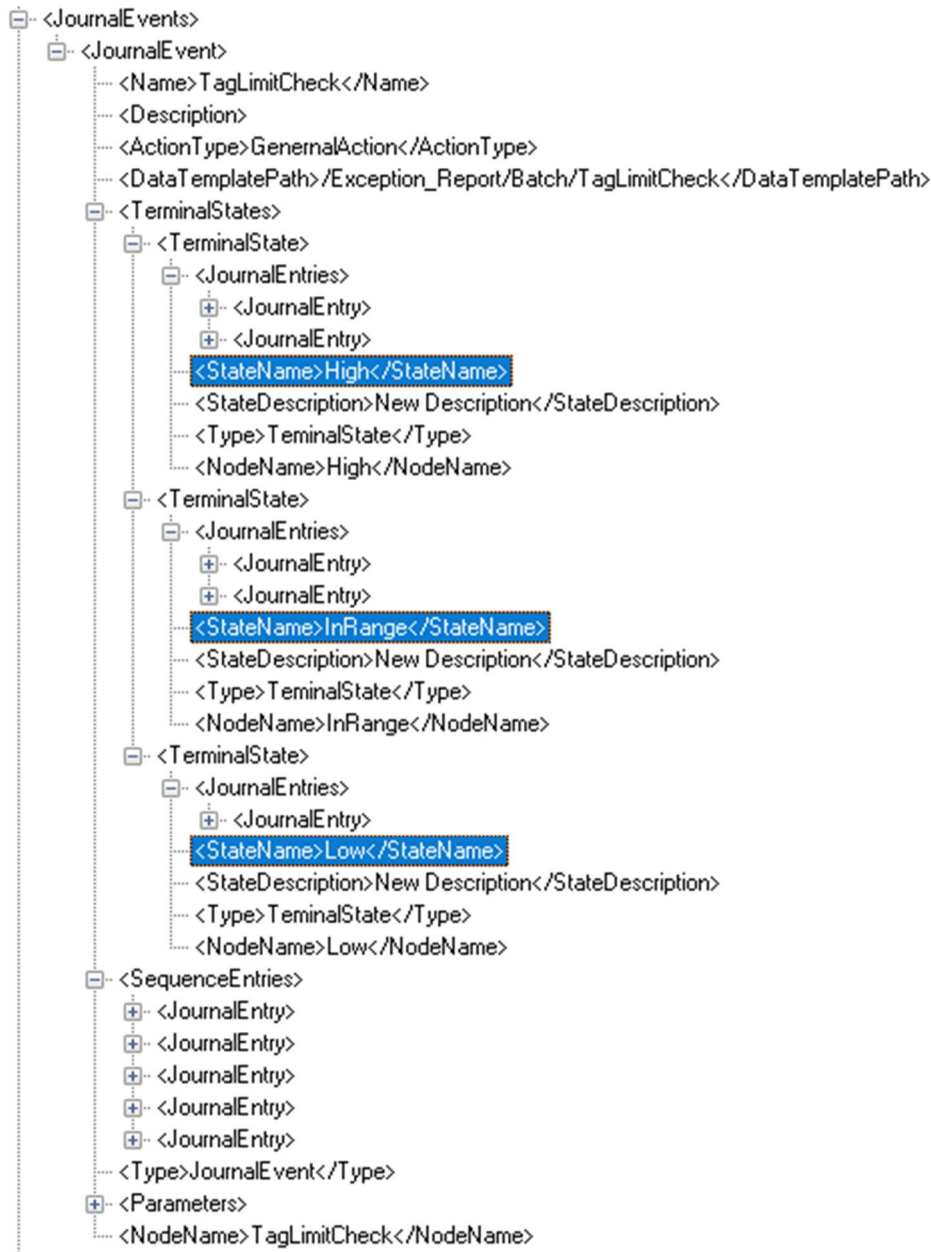
2. Journal Action Edit: switch to the Action States tab ► + ► State Name: InRange (Index: 1) ► + ► State Name: Low (Index: 2) ► + ► State Name: High (Index: 3)
3. Select the “InRange” (Index: 1) State ► check the “Default State” ► click the *Transition Equation* space ► type in *IF '{DataSource}' <= {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
4. Select the “InRange” State ► Result Definitions section ► + ► Name: Message ► Type: Simple ► drag InRangeMSG (left) to Value field (right)
5. Select the “Low” State ► Result Definitions section ► + ► Name: Message ► Type: Simple ► drag LowMSG (left) to Value field (right)
6. Select the “High” State ► Result Definitions section ► + ► Name: Message ► Type: Simple ► drag HighMSG (left) to Value field (right) ► + ► Name: Trend ► Type: Trend ► drag Data Source (left) to Data Source field (right) ► OK
7. You can also customize the trend to hide the legend and title, click on Properties:



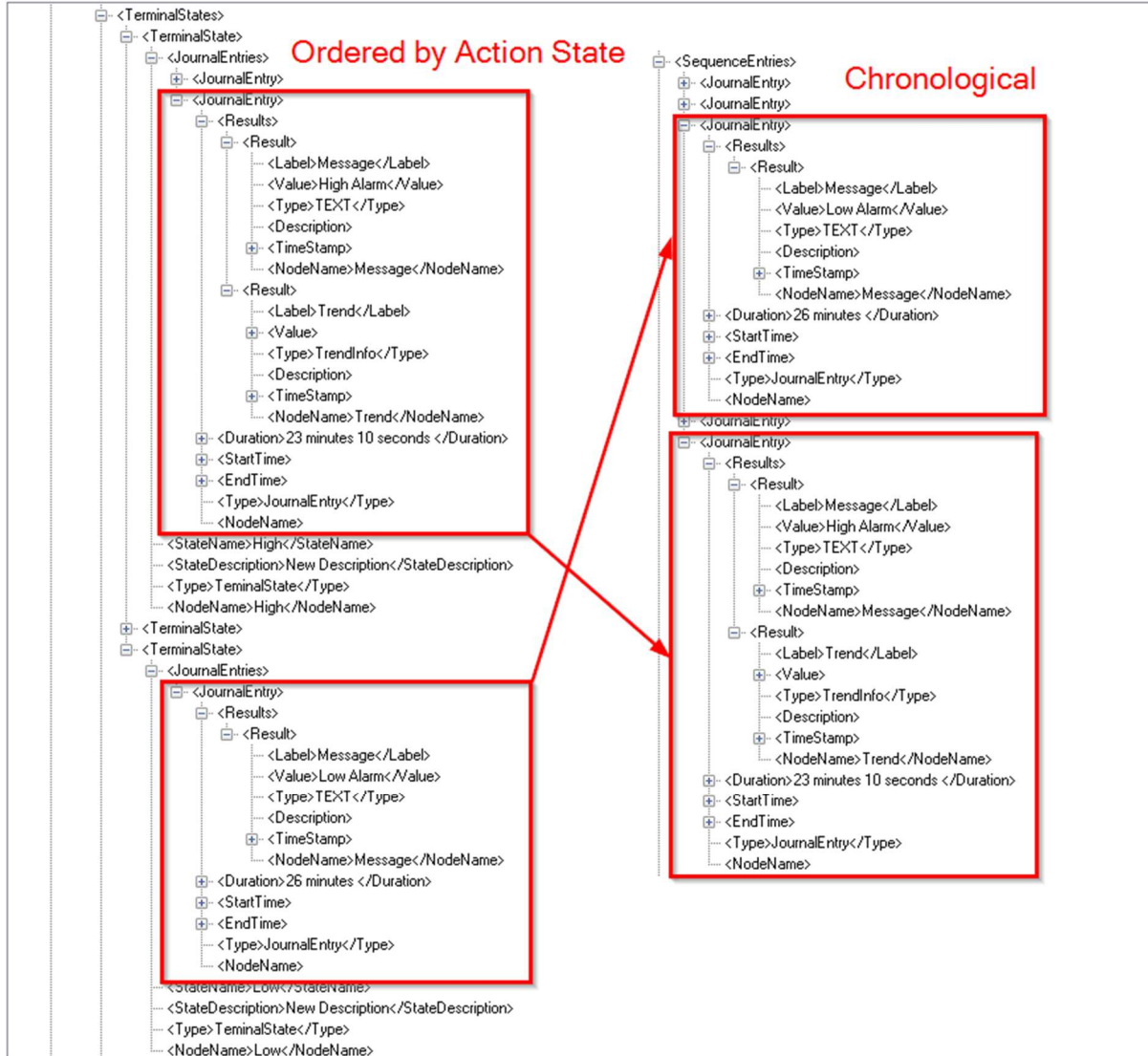
8. Save Report Template ► View ► Test Report ► Use Event Frames (or F3) ► Server: PISRV01 ► Database: THE Database... ► Search ► select a completed Procedure EF ► OK

**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 are split by each Action State



#### 4. Sequence Entries are ordered chronologically



**Format**

1. Switch to the Format Template tab
2. Create a Section ▶ select the 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 (you must search for a High JournalEntry with 2 Results)
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

State	Start Time	End Time	Trend
High Alarm	29-May-22 10:59:02 AM	29-May-22 11:33:02 AM	
In Range	29-May-22 11:33:02 AM	29-May-22 11:36:32 AM	---
Low Alarm	29-May-22 11:36:32 AM	29-May-22 12:02:32 PM	---
In Range	29-May-22 12:02:32 PM	29-May-22 12:22:02 PM	---
High Alarm	29-May-22 12:22:02 PM	29-May-22 12:45:12 PM	

### 9.3 Background: Apply Formatting 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.

#### 9.3.1 Format Template Condition Field

In the Format Template pane ► a formatting object ► Expressions section or tab ► Condition. **Note:** only one condition can exist (TRUE) in an item.

Example: a simple Journal Action that by its definition can lead to two different states: Active or Inactive. In Active state, users wish to display a trend and in Inactive state, users wish to display a simple message (example: “Unit is NOT Active”).

In this example, **users require two (2)** formatting objects:

1. For the message, Condition: `boolean(Results/Result[Label="Message"])`
2. For the trend, Condition: `boolean(Results/Result[Label="Trend"])`

**Note:** examples are representative only, users will have to apply the exact phrasing based on users’ configuration.

#### 9.3.2 Iterators: Iterator Result Condition

Example: users only want to see data of value less-than 54.

The screenshot shows a dialog box titled "Report - Table Row Edit" with a "Table Row Edit" header. It has "OK" and "Cancel" buttons and a "Help" icon. The "Expressions" tab is selected, showing a "Condition" field with the text "PIUnitName = "R-2401"". Below this is an "Iterators" section with an "Iterator" field containing "rnalEvent[Name="TagLimitCheck"]/SequenceEntries/JournalEntry" and an "Iterator Result Condition" field containing "Property[@Descriptor="220000"] < 54".

The example is **using Iterator** to display compressed data.

**Condition:** PIUnitName = "R-2401" means, users only interest in R-2401 equipment data.

**Iterator Result Condition:** Property[@Descriptor="220000"] < 54 means, display only data of value <54.

### 9.3.3 XPath Functions

Data type and <b>Function</b>	
boolean <b>starts-with</b> (string, string)	The <b>starts-with</b> function returns true if the first argument string starts with the second argument string, and otherwise returns false.
number <b>count</b> (node-set)	The <b>count</b> function returns the number of nodes in the argument node-set.
boolean <b>contains</b> (string, string)	The <b>contains</b> function returns true if the first argument string contains the second argument string, and otherwise returns false.
string <b>substring-before</b> (string, string)	The <b>substring-before</b> 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, <code>substring-before("1999/04/01", "/")</code> returns 1999.
string <b>substring-after</b> (string, string)	The <b>substring-after</b> 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, <code>substring-after("1999/04/01", "/")</code> returns 04/01, and <code>substring-after("1999/04/01", "19")</code> returns 99/04/01.
string <b>substring</b> (string, number, number?)	The <b>substring</b> 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, <code>substring("12345", 2, 3)</code> 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, <code>substring("12345", 2)</code> returns "2345".



## 9.4 Knowledge-check: Conditional Formatting

### Objectives

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

### Problem Description

In the previous report, apply conditional formatting options to change the appearance:

- Replacing the “---” symbol (indicates missing data) with “N/A” when not a High Alarm
- Removing the “In Range” state from the table

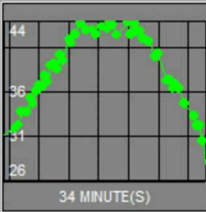

### Approach

Add a condition to show the only the Trend object when the trend object exists

Add “N/A” Text object to the same column, with the opposite condition.

Use the Iterator Result Condition field to not iterate any “In Range” message

**Report Preview:** (notice: no “In Range” State)

State	Start Time	End Time	Trend
High Alarm	29-May-22 10:59:02 AM	29-May-22 11:33:02 AM	
Low Alarm	29-May-22 11:36:32 AM	29-May-22 12:02:32 PM	N/A
High Alarm	29-May-22 12:22:02 PM	29-May-22 12:45:12 PM	

### Step by step solution:

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

1. Login to RtReports Editor
2. Open the *Exception\_Report* template from the previous exercise
3. In the Format Template tab ► expand the Section ► expand the 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

The screenshot shows the 'Report' tab in the software interface. On the left is a tree view of the report structure, including sections like 'Section', 'Entry', 'Alarms', and 'Values'. The main area displays a table of alarm events with columns for State, Start Time, End Time, and Trend. Each 'High Alarm' row includes a trend graph showing a peak in value over time.

State	Start Time	End Time	Trend
High Alarm	29-May-22 10:59:02 AM	29-May-22 11:33:02 AM	
In Range	29-May-22 11:33:02 AM	29-May-22 11:36:32 AM	N/A
Low Alarm	29-May-22 11:36:32 AM	29-May-22 12:02:32 PM	N/A
In Range	29-May-22 12:02:32 PM	29-May-22 12:22:02 PM	N/A
High Alarm	29-May-22 12:22:02 PM	29-May-22 12:45:12 PM	

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"*

The screenshot shows the 'Report - Table Row Edit' dialog box. The 'Expressions' tab is selected, and the 'Iterator Result Condition' field contains the expression: *Results/Result[Label="Message"]/Value != "In Range"*. Other fields include 'Condition' (empty), 'Iterator' (BatchJournal/JournalEvents/JournalEvent[Name="TagLimitCheck"], and 'Condition' (empty).

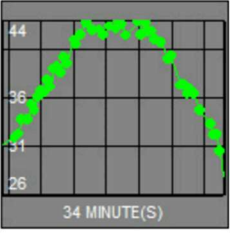
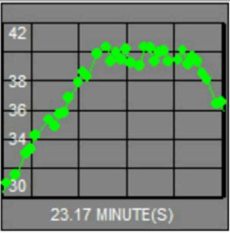
3. Save Report Template ► Refresh or F5

**Format Template**

- [-] Exception\_Report
  - [-] CoverSheet
  - [-] ReportHeader
  - [-] PageHeader
  - [-] Section
    - [-] SectionHeader
      - [-] Entry
        - [-] Alarms
          - [-] Values
            - [-] State
            - [-] StartTime
            - [-] EndTime
            - [-] Trend
- [-] SectionFooter
- [-] PageFooter
- [-] ReportFooter

**Report**

Section Begin

State	Start Time	End Time	Trend
High Alarm	29-May-22 10:59:02 AM	29-May-22 11:33:02 AM	
Low Alarm	29-May-22 11:36:32 AM	29-May-22 12:02:32 PM	N/A
High Alarm	29-May-22 12:22:02 PM	29-May-22 12:45:12 PM	

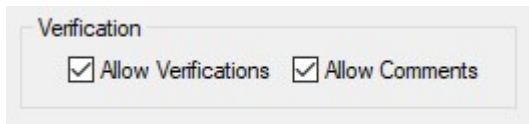
## 10. (Optional) Advanced Topics for RtReports Editor

### 10.1 Background: Format Template Options

The following options are available in the Format Template pane, right-clicking and selecting Property of the root object.

Comments, Verifications and Printing have system-wide switches found on the RtReports server setting files (%PIHOME%\RtReports\web.config, see section on RtReports Installation for details).

#### 10.1.1 Disable Comments and Verifications



The system-wide default is to allow comments and verifications on all report sections (when users have the appropriate permissions).

Under the **Format Root** tab, it is possible to disable Comments and Verifications for the entire report.

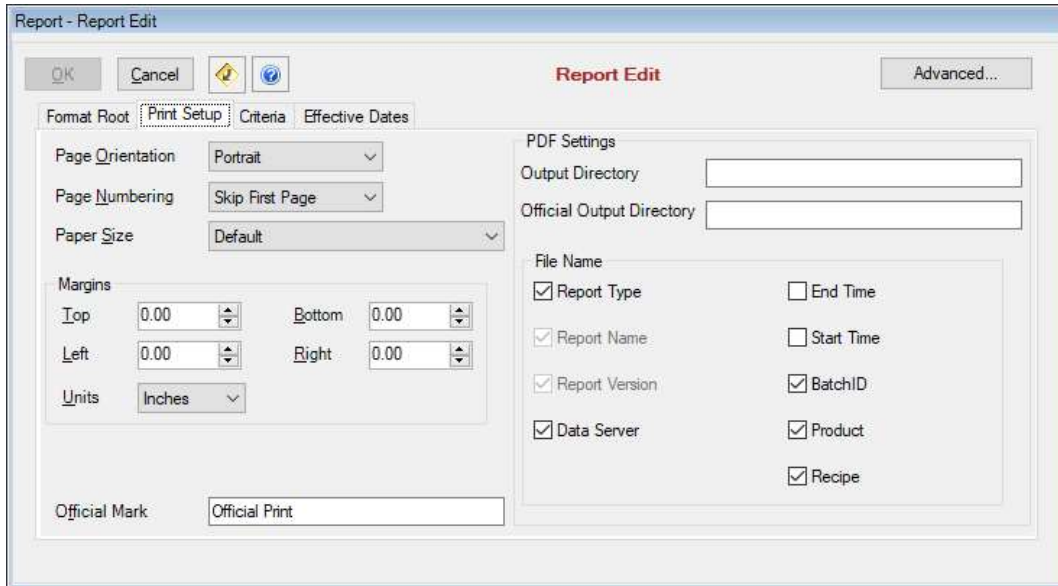
Note that it is also possible to disable commenting and verifying for each section, via the specific Properties options.

#### 10.1.2 Printing

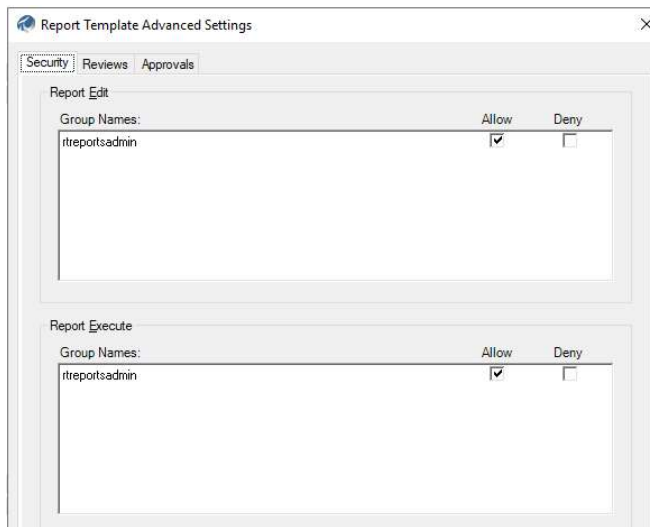
From the Print Setup, tab you can customize the following:

- Page Orientation, Numbering and Paper Size
- PDF Margins
- Text for the official watermark (after the report has been approved)
- Output directories for un-official and official PDF files

- File Name structure: Report Type (Batch/Time), Data Server, BatchID, Product, Recipe, Start and End Time



### 10.1.3 Security, Reviews and Approvals



Via the Report Edit’s **Advanced...** button, the following settings should be set up prior to releasing the report:

**Security:** Add groups that can edit the report (find it in RtReports Editor) and/or execute the report (find it in RtReports Generator)

**Reviews:** Add Reviewing Steps and define Reviewing Groups

**Approvals:** add Approval Cycles, defining Approving Group, Approving Type and Approval Role (custom fields)

## 10.2 Background: Finishing a Report Template

### 10.2.1 Report Template Status

All Report Templates must have a Version number and Status. An initial new Report Template starts at Version 1.0 and Status “In Edit”, through the last Review Cycle, the Status will be automatically changed to “Released”.

**Only one released version of a Report Template may be run against an official batch or time context.**

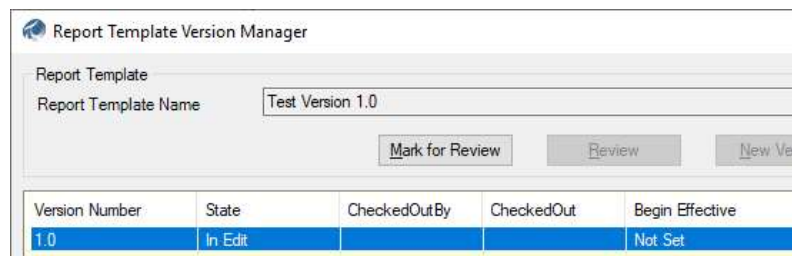
Status	
In Edit	The Report Template is in development (edit)
In Review Step #	The Report Template is ready for (or in) Step # review
Released	The Report Template is “Released” to Production

possible Report Version Status List

### 10.2.2 Releasing a Report Template

To release a Report Template, the Author must first mark it for review and notify the first reviewing groups; once all reviewing steps are completed, the Report Template Status will be set to “Released” automatically. Rejection in any Review Step will set the Template Status back to “In Edit”.

To mark a Report Template for Review, the author of the Report Template must be logged to RtReports Editor, then on Menu Tools ► Report Version Manager... ► Mark for Review



The author should now notify the Reviewing Groups that the Report Template is ready for Review. If a Report Template has **NO Reviewing Step**, the author can **Review** and release the Report Template immediately.

The **Reviewer** of a Report Template now can login to RtReports Editor, open and verify the content of the Report Template. To release the report:

- On Menu Tools ► Report Version Manager... ► **Review** ► enter Comments ► Accept or Reject.

**Notice** the number of Reviewing Step, Status: “In Review Step #”

- Click the “*Template Reviews*” (bottom) to view all the reviewing steps.

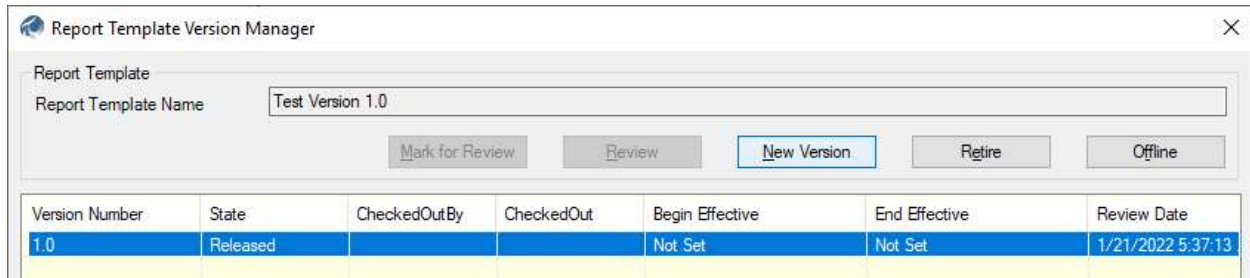
### 10.2.3 New Version for a Report Template

A Report Template of Status “Released” cannot be changed. To continue working with this report, the RtReports Editor user must create a new version and then make changes to the new version.

A user can only start a new version of a “Released” Report Template; the version is automatically incremented by 1.

After logging in to RtReports Editor, open the desire report template (latest version)

- On Menu Tools ► Report Versioning Manager... ► New Version...

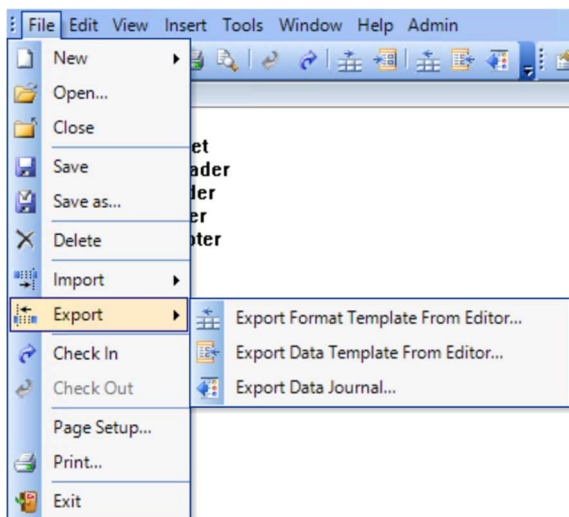


## 10.3 Background: Importing and Exporting

RtReports provides users with export and import utilities for Report Template, Data Toolbox: Predefined, and Format Toolbox: Style Sheet. The exported data is in .xml file. The exported .xml file may be shared among company sites, as a backup... **Note:** importing a Report Template .xml file (backup) is copying the Report Template to create new one.

### 10.3.1 Import/Export a Report Template

Each Report Template consist of a Data Template and a Format Template that users must import and/or export each independently for a complete Report Template.

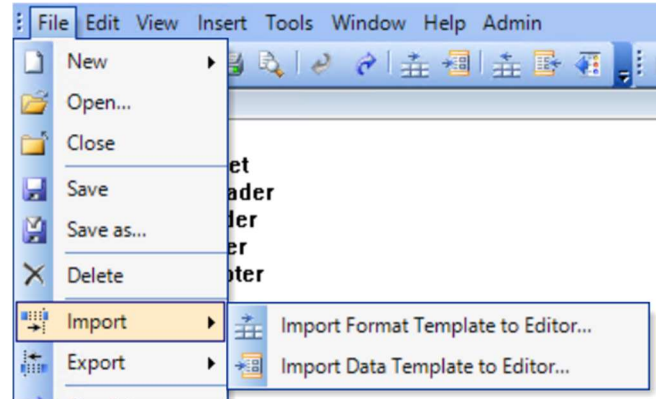


Steps to Export a Report Template:

- Login to RtReports Editor
- Open the Report Template
- File ► Export ► Export **Format Template** From Editor...
- File ► Export ► Export **Data Template** From Editor...

## Steps to Import a Report Template:

- Login to RtReports Editor
- Create a new empty Report Template of any type (Batch or Time, it will be reset to the imported type), or open one to be overwritten.
- File ► Import ► Import **Format Template** to Editor...
- File ► Import ► Import **Data Template** to Editor...





## 10.4 Knowledge check: Import and Release a Batch Report Template

### Objectives

Quickly create and release a Batch Report Template.  
 Using the Import function to import the Data Template and Format Template.  
 Enable/disable *Allow Comments* and/or *Allow Verifications* for a report

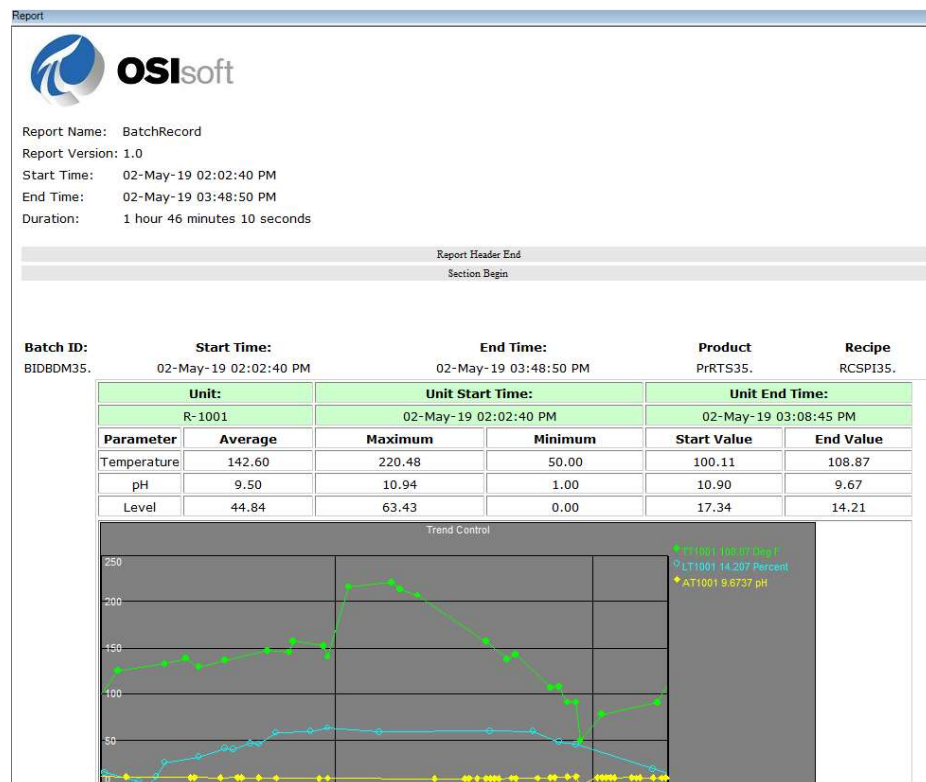
### Problem Description

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

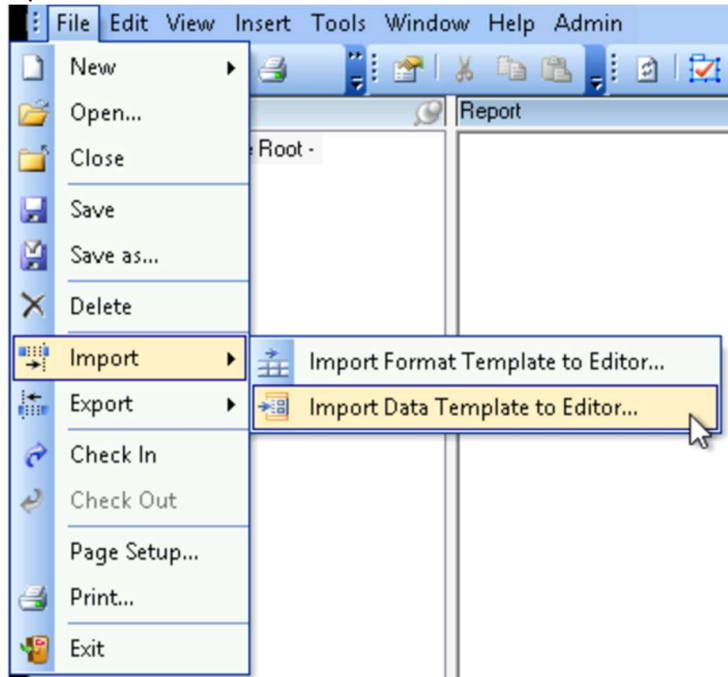
Create a new Batch Report Template called *BatchRecords*.  
 Import the AF Attributes based Data Template file C:\Class\Exercises and Solutions\BatchRecords\BatchRecords\_DataTemplate\_AFAttributes.xml  
 Import the Format Template file C:\Class\Exercises and Solutions\BatchRecords\BatchRecords\_FormatTemplate.xml.  
 Add an Approving Group and a Reviewing Step (both can be set for RtReports Admin)  
 Use the Report Version Manager under the Tools menu to release the report.

### Preview:



**Step by step solution:**

1. Login to the RtReports Editor
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\_AFAttributes.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. Switch to the Format Template pane ► double click on Root BatchRecords ► check to enable Allow Comments and Allow Comments
6. Save and test: *View* ► *Test Report* ► *Use Event Frames* (or F3) ► find and select a completed 1K Reactors EF (*Procedure*) ► and verify
7. Release the Report Template: *Tools* ► *Report Version Manager...* ► *Mark for Review* ► *Review*

## 10.5 Knowledge check: Import and Release a Time Report Template

### Objectives

Quickly create and release a Time Report Template.  
Using the Import function to import the Data Template and Format Template.

### Problem Description

Quickly create a new Time Report Template by importing the Data and Format Templates.  
Release the Time Report Template.

### Approach

Create a new Time Report Template called *DailyReport*.

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

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


Add an Approving Group and a Reviewing Step (both can be set for RtReports Admin)

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

**Note:** you cannot Generate any partial (incomplete) report

Preview:

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 <small>04-May-19 12:59:01 AM</small>	92.7772 <small>04-May-19 02:23:31 AM</small>	60.8314
Level Reactor 1	0.0000 <small>04-May-19 12:59:01 AM</small>	43.9049 <small>04-May-19 01:48:01 AM</small>	22.2191
Temperature Reactor 1	0.0000 <small>04-May-19 12:59:01 AM</small>	52.4452 <small>04-May-19 11:04:58 PM</small>	18.1636
Concentration Reactor 1	0.0000 <small>04-May-19 12:59:01 AM</small>	46.5248 <small>04-May-19 02:11:01 AM</small>	21.3580

Section End  
Section Begin

Trend Control

DEG. C
037131.97.983.000. C

**Step by step solution:**

1. Login to the RtReports Editor
2. Create a new Batch Report Template: *File* ► *New* ► *Batch Report Template* ► Name: DailyReport ► OK
3. Import the Data Template XML file: *File* ► *Import* ► *Import Data Template to Editor...* ► browse "C:\Class\Exercises and Solutions\DailyReport\DailyReport\_DataTemplate\_PITags.xml" ► Open
4. Import the Data Template XML file: *File* ► *Import* ► *Import Format Template to Editor...* ► browse "C:\Class\Exercises and Solutions\DailyReport\DailyReport\_FormatTemplate.xml" ► Open
5. Save and test: *View* ► *Test Report* (or F2)

The screenshot shows a dialog box titled "Time Based Report". It has a close button (X) in the top right corner. The dialog is divided into two main sections: "Server" and "Context".

**Server Section:**

- Data Server:** A dropdown menu with "PISRV01" selected.
- Asset Server:** A dropdown menu with "{None Selected}" selected.
- Asset Database:** A dropdown menu with "{None Selected}" selected.

**Context Section:**

- Start Time:** A dropdown menu with "\*-1d" selected, accompanied by a calendar icon.
- End Time:** A dropdown menu with "|" selected, accompanied by a calendar icon.

On the right side of the dialog, there are three buttons: "OK", "Cancel", and "Help".

6. Release the Report Template: *Tools* ► *Report Version Manager...* ► *Mark for Review* ► *Review*

## 10.6 Background: Report Parameters

Report Parameters (sometimes referred to as Runtime Parameters) feature is introduced starting in RtReports version 3.2.

The Report Parameter allows user to make changes to its value at runtime in RtReports Generator. This feature allows for powerful ad hoc reporting, troubleshooting...

The three (3) Value Data Type are String, Numeric, or List. **Data Type: List** is best to avoid any incorrect data entry).

Example: a list of PI Tags (of Data Type: String), or a list of multiplication factors (Data Type: Numeric)

### Implementation steps for a Report Parameter on related equipment measurements

(e.g., pH, Tank Level, Temperature)

- Create or open an existing Report Template
- Go to Menu Tools ► Report Parameters
- Click on the '+' sign ► change the name (e.g., Param0), add a **Prompt** message
- Specify Data Type and Default Value
- For a List Data Type, click on ... to configure possible values
  - Only PI Tag names and legacy module/batch data (Alias/Property/Module) can be currently searched
  - For AF Elements and Attributes, manual input is needed
- For Batch Reports, create a Unit Template and add a Journal Action under it
  - Add an Action Parameter (click on +)
  - Select the Type (PIAlias, PIProperty or AFAttribute)
  - Hit the Cancel button
  - For the Action Parameter Value, change it to the given report parameter name in square brackets (e.g. [Param0])

## 10.7 Knowledge check: Create and Release a Report with Parameters

### Objectives

Create a Batch Report Template to report Summary Value of data (Element Attributes) that are selected by the user at runtime.

### Problem Description

Users want a Batch-based report to analyze a process data point of their selection (show Minimum, Maximum, and Average) at runtime of either “pH”, “Level” or “Temp”.

### Approach

Create a new Batch Report “Runtime Parameters Batch Report”.

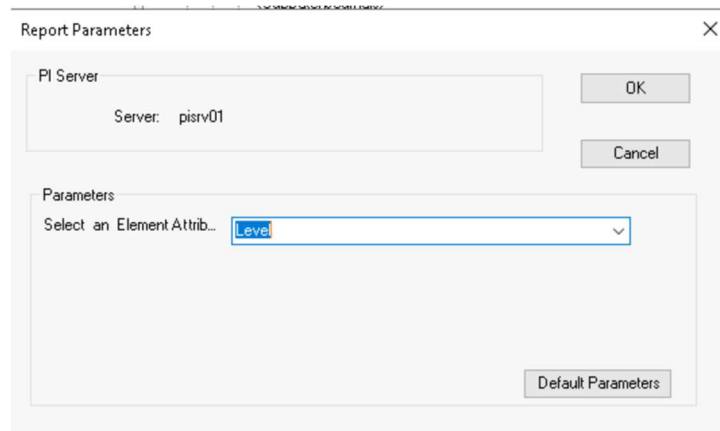
Configure an “Attributes” Parameter List that consists of the values Level, pH, and Temp.

Insert a Unit Journal Action “SummaryJournalAction” of Result Definitions type “Summary Value” of Minimum, Maximum, Average, and Range of the selected Attribute.

Display the Result Definitions of the SummaryJournalAction using the Trace Set object

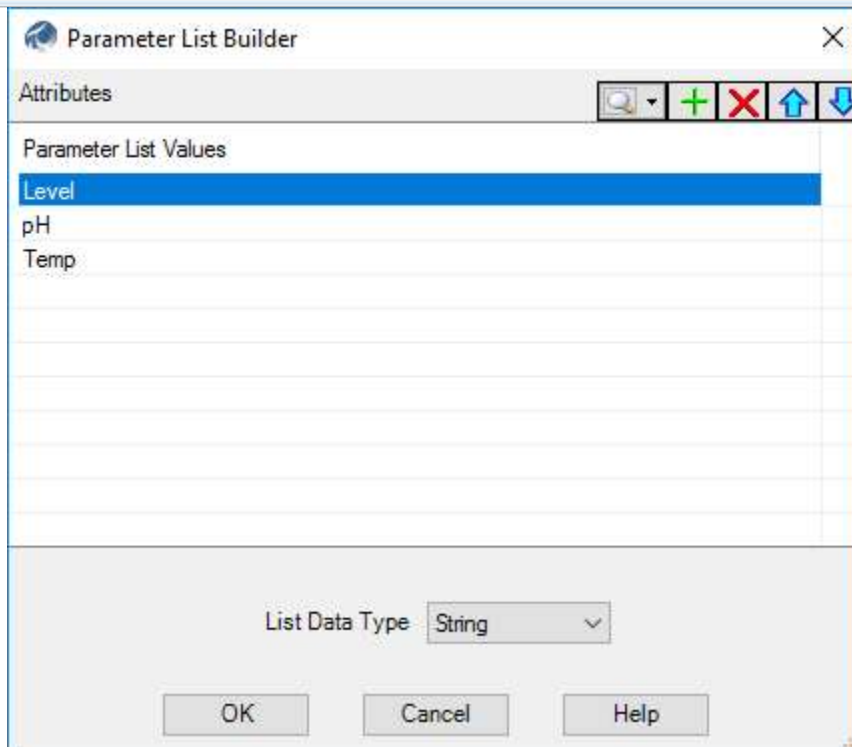
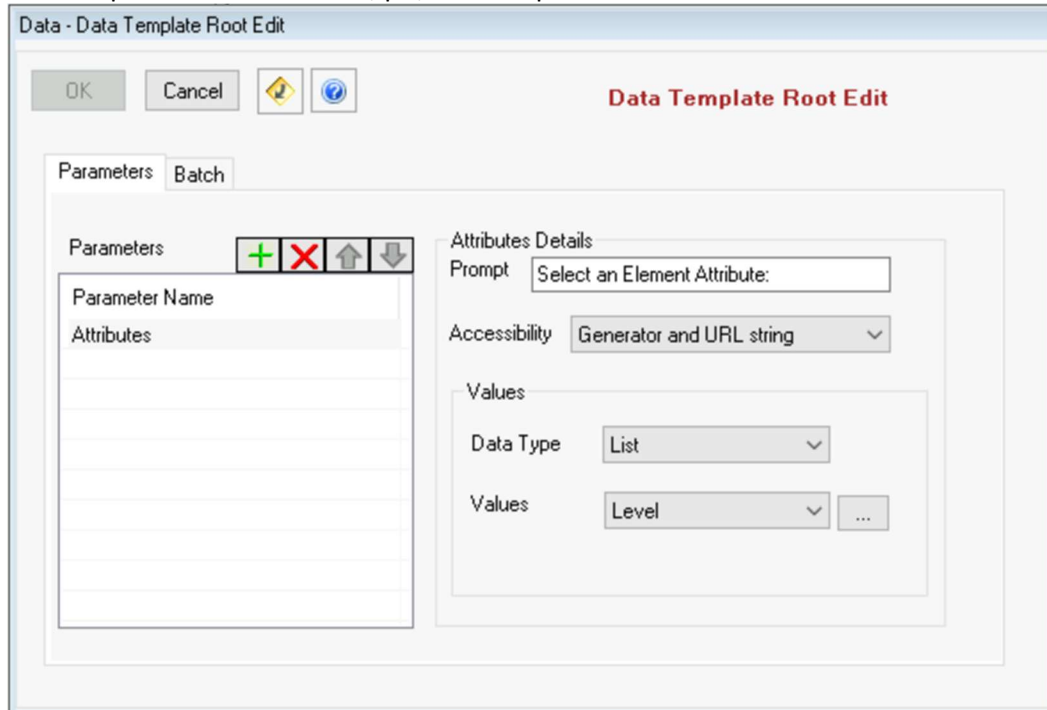
Test the report against a Batch/EF of the 1K Reactors by selecting an Attribute and then click Refresh.

**Testing on the Editor will yield this dialog box after selecting batch context:**



**Step by step solution:**

1. Login to RtReports Editor
2. File ► New ► Batch Report Template... ► Name: Runtime Parameters Batch Report
3. Add the “Attributes” Parameter: Menu Tools ► Report Parameters ► + ► call it “Attributes” ► Prompt: “Select an Element Attribute: “ ► Data Type = List ► click ... (to the right of Values) ► click + 3 times ► replace them with Level, pH, and Temp ► OK



4. Click OK to close the Report Parameters configuration



5. Switch to the Data Template pane
6. Select *Batch* ► Insert ► Unit Template ► OK
7. Select the Unit Template ► Insert ► Journal Action ► Name: SummaryJournalAction
8. Action Parameters: + ► Name: Attr ► Type: AFAttribute ► Cancel the Attribute Search prompt ► Value: [Attributes]
9. Journal Action Edit: Action Results tab
10. Result Definitions: + ► Name: Summ ► Type: Summary Value ► drag Attr to Data Source ► check Yes for Minimum, Maximum, and Average ► OK

**Journal Action Edit**

Event Name: Direct Action

Output	Name	Type	Description
<input type="checkbox"/>	Summ	Summary Value	New Description

**Action Parameters**

Name	Type	Value	Dynamic
Attr	AFAttribute	[Attributes]	Yes

**Time Parameters**

Start Time Context: Context Start

End Time Context: Context End

**Data Parameters**

Data Source: [[Attr]]

Display Digits: [ ]

Conversion Factor: [ ]

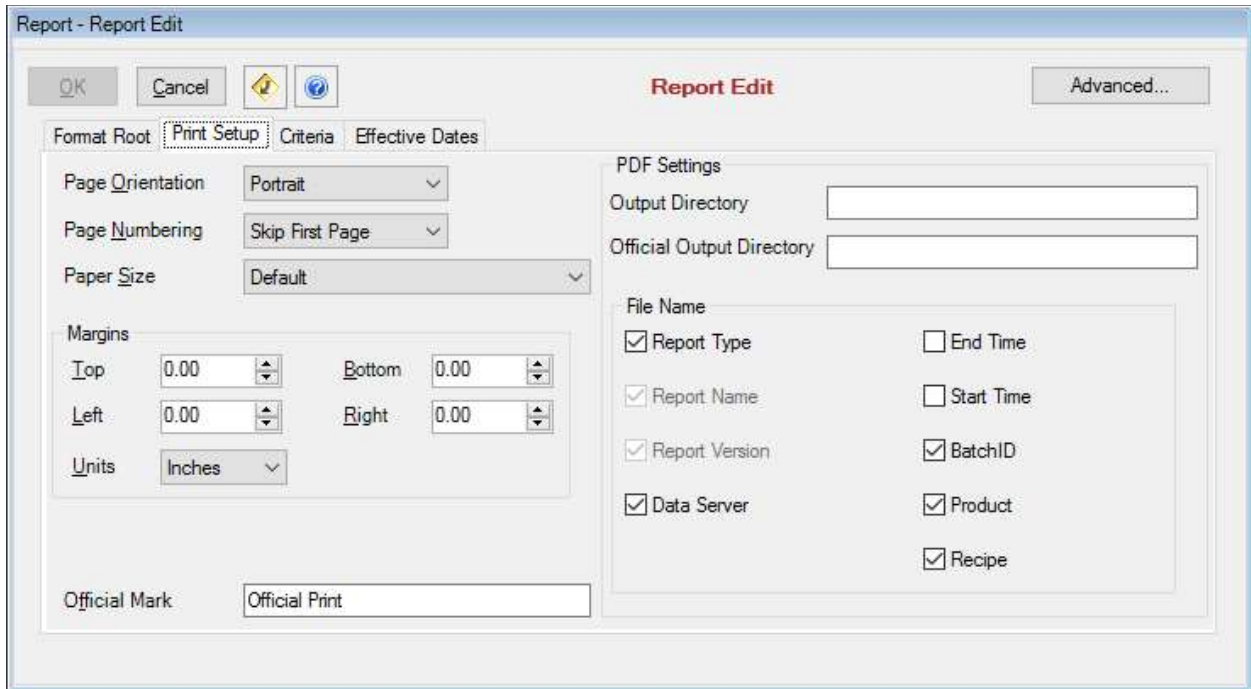
Minimum:  Yes  No

Maximum:  Yes  No

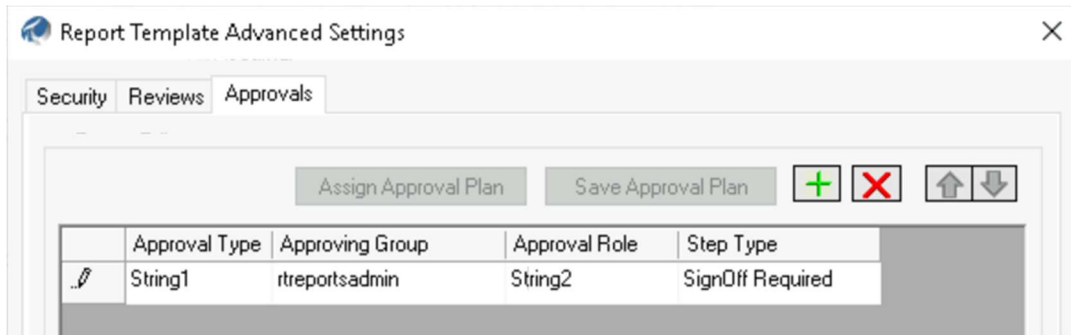
11. Save Report Template ► View ► Test Report ► Use Event Frames (or F3) ► search a completed Procedure of 1K Reactors...
12. Switch to the Format Template pane
13. Select root ► Insert Section
14. Expand Section ► select Entry ► Properties ► Expressions tab ► browse and drag BatchJournal/UnitBatchJournals/UnitBatchJournal to Iterator field
15. Select Entry ► Insert ► Trace Set ► browse and drag *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 ► X columns Time, ErrorCode, ErrorDescription, and IsGood ► OK
16. Save Report Template ► Refresh (of F5) and verify

### Verify and Release the Batch Report

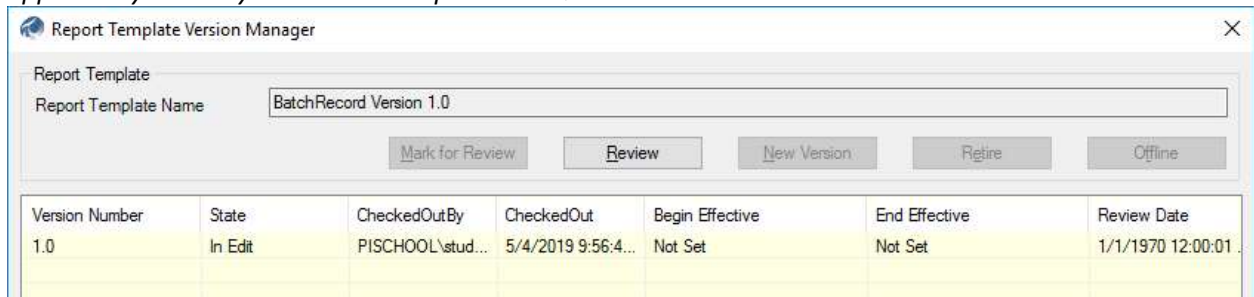
1. Switch to Format Template pane ► verify the report settings:



- Double click on the report root
- 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...* ► Mark for Review or if no Approval cycle was configured, click Review button (to start Review) ► select **Yes** to warning “*This report has no approval cycle. Do you still want to proceed?*” ► Close



3. Click *Close* to complete releasing the report

## 11. (Optional) PI Baseline Service

### 11.1 Background: Introduction

PI Baseline Services (also known as RtBaseline) 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.

The PI Baseline Services is installed as part of the RtReports Server installation. All configuration is performed through a web browser URL: **Error! Hyperlink reference not valid..**

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



Configure data sources and datasets using the links below.

**Data Sources**  
Data sources contain information on how to connect to sources of data used in displays.

[Relational Data Sources](#)

**Datasets**  
Datasets describe queries that return data used in displays.

[Relational Datasets](#)

## 11.2 Background: 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 2+ PI Trust: <RtReports Server IP> + 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.

Steps to create a Data Source:

- Open the PI Baseline Service Administrator page
- Click [Relational Data Sources](#) hyperlink
- Click New ► Define Data Source Name: ► Description: ► Database Type: (select from dropdown) ► ...
- Define the Connection String... May need to click Edit Connection String button
- Click Save
- Click Test Connection (must get "Test Successful" to continue)

The screenshot shows the configuration interface for a Relational Data Source. On the left, there are labels for 'Data Source Name', 'Description', 'Database Type', 'Data Source Name', 'Source Time Zone', 'Password', and 'Connection String'. On the right, there are input fields and dropdown menus. The 'The DSN' field is empty. The 'Database Type' dropdown is set to 'SQL Server'. The 'Source Time Zone' dropdown is set to '(UTC) Coordinated Universal Time'. There is a checked checkbox for 'No Password'. The 'Connection String' field contains the text: 'Server=PISRV01;Database=RTBaseline;Trusted\_Connection=true;'. At the bottom right, there are three buttons: 'Edit Connection String', 'Test Connection', and 'Save'.

### 11.3 Background: Datasets

Relational Datasets are the SQL queries built based on the defined Data Source. To create a new Dataset:

- Click on hyperlink **Datasets** ► [Relational](#)
- Click New ► Define Data Set Name, Description and Data Source
- Define the SQL Statement
- Click on Verify SQL button to allow defining Placeholders accordingly
- Once Placeholders have Display Name, Default Value, Type/Format, click Verify SQL button again

- Save
- Click Preview button to preview the SELECT result

Preview

Timestamp	Temperature
04-Feb-2022 19:48:56	14.3243
04-Feb-2022 19:54:56	14.2888
04-Feb-2022 20:00:56	14.4043
04-Feb-2022 20:06:56	14.8779
04-Feb-2022 20:12:56	14.8162
04-Feb-2022 20:18:56	15.2182
04-Feb-2022 20:24:56	15.2694
04-Feb-2022 20:30:56	15.3447
04-Feb-2022 20:36:56	15.388
04-Feb-2022 20:42:56	15.4414

1 2 3 4 5 6 7 8

## 11.4 Knowledge check: Incorporate External Data in Report

### Objectives

Discover how to bring external data (non-PI Data) from a relational data source into a report.

### Problem Description

Need a Bay Daily Temperatures Report.

### Approach

Configure a Time Report Template “BayDailyTemperatures”

Configure a Data Source: Bay Temperature – SQL Server: PISRV01 and Database: RtBaseline

Configure a Dataset: Bay Temperature – display Timestamp vs. Temperature serial data

Configure a table to display the Timestamp vs. Temperature serial data

The screenshot displays the 'nat Template' configuration window for a report titled 'Bay Daily Temperatures'. The left pane shows a hierarchical tree structure of the report template components, including CoverSheet, ReportHeader, PageHeader, Section, SectionHeader, Entry, Table, Header, Timestamp, Temperature, Data, Time UTC, SectionFooter, PageFooter, and ReportFooter. The 'Table' component is expanded to show its 'Header' and 'Data' sections. The 'Header' section contains 'Timestamp' and 'Temperature' fields. The 'Data' section contains 'Timestamp', 'Time UTC', and 'Temperature' fields. The right pane shows the rendered report output, which is a table with two columns: 'Timestamp' and 'Temperature'. The table contains 15 rows of data, showing timestamps from 28-May-22 06:18:43 PM to 28-May-22 07:30:43 PM and corresponding temperature values ranging from 9.1887 to 9.2288.

Timestamp	Temperature
28-May-22 06:18:43 PM	9.1955
28-May-22 06:24:43 PM	9.1965
28-May-22 06:30:43 PM	9.2015
28-May-22 06:36:43 PM	9.2109
28-May-22 06:42:43 PM	9.2216
28-May-22 06:48:43 PM	9.2526
28-May-22 06:54:43 PM	9.246
28-May-22 07:00:43 PM	9.2274
28-May-22 07:06:43 PM	9.2099
28-May-22 07:12:43 PM	9.1947
28-May-22 07:18:43 PM	9.1887
28-May-22 07:24:43 PM	9.209
28-May-22 07:30:43 PM	9.2288

### Step by step solution:

#### Create the Data Source and Dataset

1. Login to the PI Baseline Services Administration page, URL: <https://PITRT01/RtReports/Admin>
2. Create a new Relational Data Source:
  - Click “Relational Data Sources” or “Relational” under Data Sources
  - Click New
  - Data Source Name: Bay Temperature
  - Database Type: SQL Server (select from the dropdown arrow)
  - Click Edit Connection String ► check Trusted Connection ► OK ► verify Connection String: Server=PISRV01;Database=RtBaseline;Trusted\_Connection=true; ► Save

- Click “Test Connection” (should get Test Successful) ► OK
3. Create a new Dataset using the SF Bay Temperature Data source:
    - Click “Relational Datasets” or “Relational” under Datasets
    - Click New
    - Data Set Name: Bay Temperature
    - Data Source: Bay Temperature
    - SQL Statement for testing: SELECT Temperature, Timestamp FROM [dbo].[BayWaterTemperature] WHERE Timestamp BETWEEN ? AND ?
    - Click Verify SQL button ► fill in the Place Holders:

Name	Display Name	Default Value	Type	Data Format
Place Holder 1	Start Time	*-8h	StartTime	DateTime
Place Holder 2	End Time	*	EndTime	DateTime

- Save
- Click Verify SQL button again
- Click “Preview” to view/verify data

#### Configure SF Bay Temperature Time Report Template

1. Login to RtReports Editor
2. File ► New ► Time Report Template...
3. Name: BayDailyTemperatures ► Recurrence: Every weekday ► Reference Start: 12:00 AM ► Reference End: 12:00 AM ► OK
4. Data Template pane ► right click Time ► Insert ► Journal Action ► Name: External Data ► Action Results tab ► + ► Result Definitions Name: Bay Temperature ► Type: *RtBaseline Services DataSet* ► select Bay Temperature ► OK ► OK

**Data - Journal Action Edit**

OK Cancel [Help] [Refresh]

Name: External Data  
Description: [Text Area]

**Action Parameters**

Name	Type	Value	Dynamic

**Journal Action Edit**

Action Options | Action States | Action Results

Event Name: Direct Action

**Result Definitions**

Output	Name	Type	Description
<input type="checkbox"/>	Bay Temperatures	RtBaseline Services DataSet	New Description

**Time Parameters**

Start Time  
Context: Context Start  
Offset: [Text Box]

End Time  
Context: Context End  
Offset: [Text Box]

**Data Parameters**

Data Source: SF Bay Temperature  
Mode: ModeEvents

5. View ► Test Report... (or F2)
6. Data Server: PISRV01 ► Assert Server: PISRV01 ► Asset Database: THE Database ► Start Time: \*-8h ► End Time: \* ► OK
7. Format Template pane ► right click root ► Insert ► Section ► OK
8. Expand Section ► right click Entry ► Insert ► Table ► Rows: 1 ► Columns: 2 ► Border tab ► Frame: Border ► Border Width: 1 ► OK
9. Label Header row and Data row ► label column 1: Timestamp ► label column 2: Temperature
10. Header row: column 1 ► Insert ► Text: Timestamp ► column 2 ► Insert ► Text: Temperature
11. Right click Data row ► Property ► Expressions tab ► browse and drag *TimeBasedRootJournal/JournalEvents/JournalEvent[Name="External Data"]/TerminalStates/TerminalState[StateName="Direct\_Action"]/JournalEntries/JournalEntry[position=1]/Results/Result[Label="Bay\_Temperatures"]/Value/TraceSet[@Name="0"]/Events/Properties* to Iterator field ► OK
12. Data row, Timestamp column ► Insert ► UTC Time ► browse and drag *Property[@Descriptor="100"]* to Source field ► OK
13. Data row, Temperature column ► Insert ► Data ► browse and drag *Property[@Descriptor="22000"]* to Source field ► OK



The screenshot displays the AVEVA PI RtReports 4.1 Administration interface, split into two main panels: 'Format Template' on the left and 'Journal' on the right.

**Format Template Panel:** This panel shows a hierarchical tree structure for a report template named 'BayDaily Temperatures'. The structure includes:

- BayDaily Temperatures
  - CoverSheet
  - ReportHeader
    - PageHeader
  - Section
    - SectionHeader
      - Entry
        - Table
          - Header
            - Timestamp
            - Temperature
          - Data
            - Timestamp
            - Time UTC
            - Temperature
            - Data
- SectionFooter (F7)
- PageFooter (F7)
- ReportFooter (F7)

```

<TimeBasedRootJournal>
  <OfficialTimeUsed>false</OfficialTimeUsed>
  <Cacheable>false</Cacheable>
  <DefaultReportParameters>
  <ReportParameters>
  <EditorVersion>4.1.4.0</EditorVersion>
  <NodeName>BayDailyTemperatures</NodeName>
  <Journals>
  <Empty>false</Empty>
  <SampleData>false</SampleData>
  <Author>PISCCHOOL\student01</Author>
  <Version>1.0</Version>
  <ReleaseDate>
  <CreationDate>
  <GenerationDate>
  <TotalSeconds>0.3439097</TotalSeconds>
  <TimeAFServer>PISRVD1</TimeAFServer>
  <TimeAFDatabase>THE Database</TimeAFDatabase>
  <JournalEvents>
  <JournalEvent>
    <Name>External Data</Name>
    <Description>
    <ActionType>GeneralAction</ActionType>
    <DataTemplatePath>/BayDailyTemperatures/Time/External Data</DataTemplatePath>
    <TerminalStates>
      <TerminalState>
        <JournalEntries>
          <JournalEntry>
            <Results>
              <Result>
                <Label>Bay Temperatures</Label>
                <Value>OSISoft.PI.Web.Services.PIDataAccess.TraceSets</Value>
                <TraceSet>
                  <Name>0</Name>
                  <QuerySpecName>0</QuerySpecName>
                  <Properties>
                  <Columns>
                  <Events>
                    <Properties>
                      <PropertyDescriptor=101>1</Property>
                      <PropertyDescriptor=100>1644007736</Property>
                      <PropertyDescriptor=22000>15.4957</Property>
                    </Properties>
                  </Properties>
                </Properties>
              </Result>
            </Results>
          </JournalEntry>
        </JournalEntries>
      </TerminalState>
    </TerminalStates>
  </JournalEvent>
  </JournalEvents>
  </TimeBasedRootJournal>
  
```

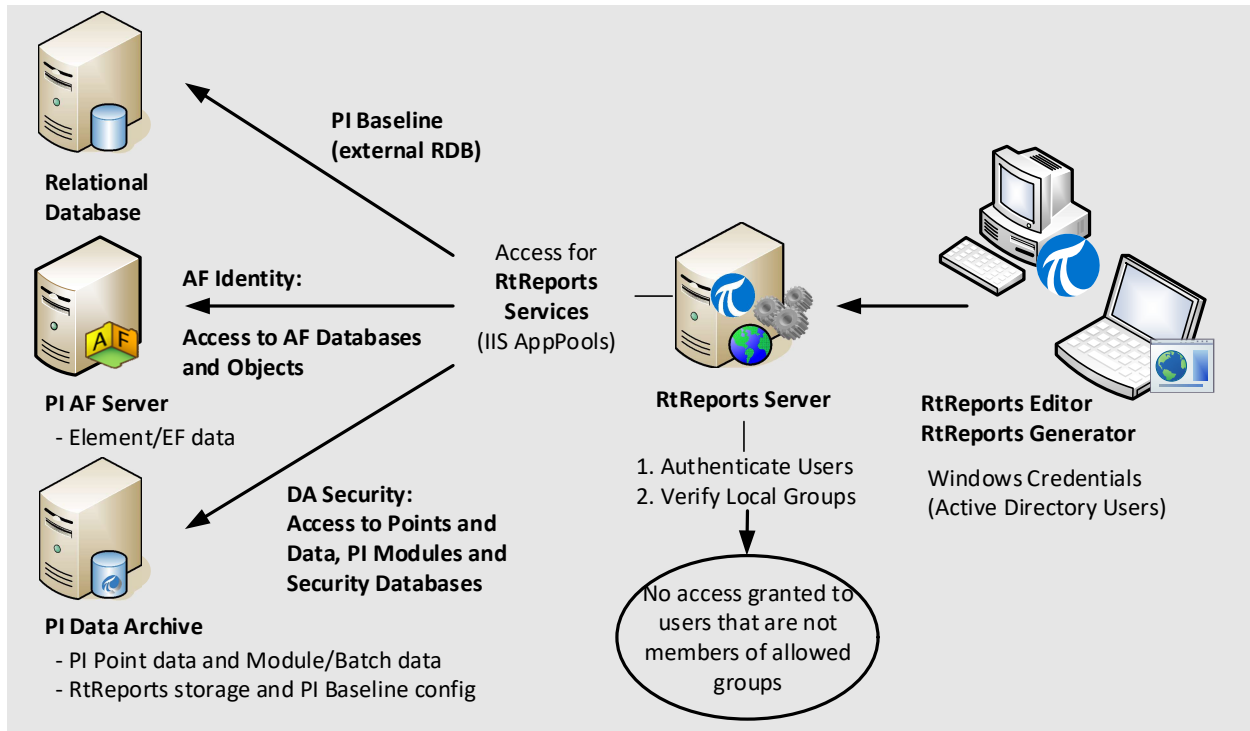
14. View Refresh (or F5)

## 12. (Optional) RtReports Security

### 12.1 Background: Introduction

RtReports benefits from Windows Credentials to validate logging users and their permissions based on local groups assigned - both on RtReports Editor and RtReports Generator.

Only the members of groups that are allowed for a report (Report Edit) may log into RtReports Editor, and only those who are allowed for a report (Report Execution) may log into RtReports Generator.



Read Access to external Data sources, PI Data Archive and PI Asset Framework data should be configured during installation for the RtReports accounts. RtReports uses the PI Data Archive to store Report Templates, PI Baseline configuration, and all information created from the report generation workflow.

### 12.2 Background: RtReports Windows Groups

The permissions that can be defined globally in RtReports are the main tasks: Print, OfficialPrint, Approve, Comments, Verification, User, Administrator, Edit, and Review.

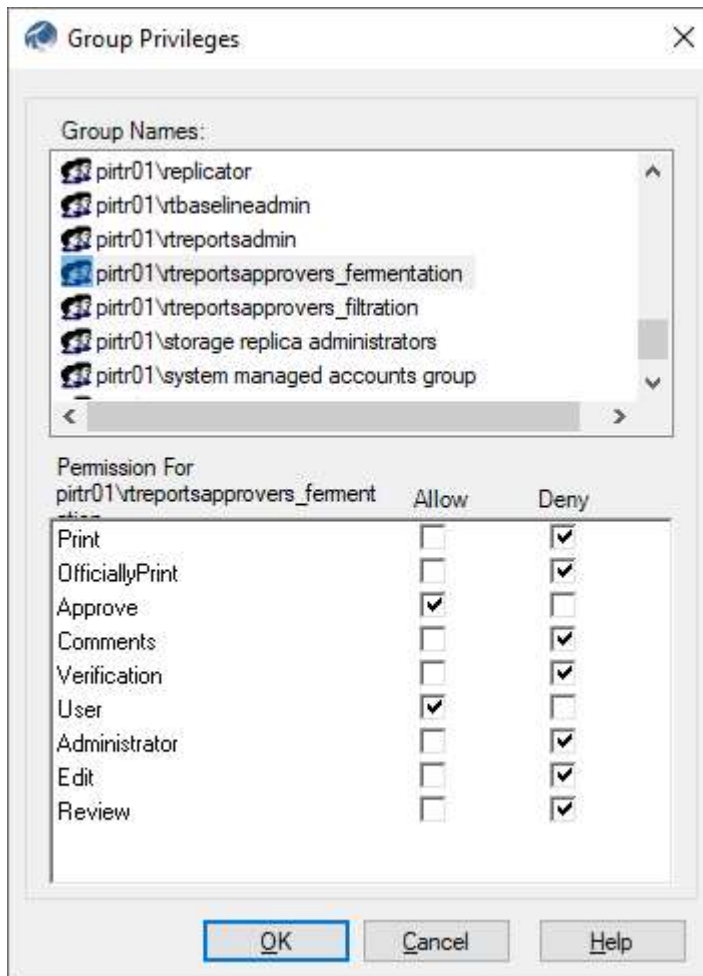
Multiple groups may be required for each task (depending on workflow). For example:

- Only Fermentation personnel can approve a Fermentation Report. Group name: RtReportApprovers\_Fermentation
- Only Filtration personnel can approve a Filtration Report. Group name: RtReportsApprovers\_Filtration

Local User Groups can be created on the server, to allow RtReports users to Edit, Review, Comment, Verify, Approve and Print. Windows Domain Groups and AD Users can be assigned to each local group.

### 12.3 Background: Grant Permissions to Groups

- The RtReports Administrator logs into RtReports Editor
- Menu Admin ► Group Privileges... ► select a Group Name ► grant Permissions by checking “Allow” boxes.



Note: The User permission is granted automatically when allowing any other permission.

## 12.4 Background: Report Template Security

### 12.4.1 Report Edit and Report Execute

**Report Edit** is a report-specific permission that can be granted for RtReports Editor use, which applies to Edit and Review permissions.

**Report Execute** is a report-specific permission that can be granted for RtReports Generator use, which applies to Print, Officially Print, Approve, Comments, Verification permissions.

### 12.4.2 Grant RtReports Windows Groups to a Report Template

Each Report Template Author grants the RtReports Windows Groups for Report Edit and/or Report Execute by checking "Allow".

**Steps:** login to RtReports Editor ► open the Report Template ► Format Template pane ► double clicks the Root ► Advance... ► Security tab ► check "Allow" to Group name(s) in Report Edit and/or Report Execute.

The screenshot shows the "Report Template Advanced Settings" dialog box with the "Security" tab selected. It contains two sections: "Report Edit" and "Report Execute". Each section has a table with columns for "Group Names", "Allow", and "Deny".

Group Names:	Allow	Deny
rtreportsapprovers_filtration	<input type="checkbox"/>	<input checked="" type="checkbox"/>
rtreportsapprovers_fermentation	<input type="checkbox"/>	<input checked="" type="checkbox"/>
rtreportsadmin	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Group Names:	Allow	Deny
rtreportsapprovers_filtration	<input type="checkbox"/>	<input checked="" type="checkbox"/>
rtreportsapprovers_fermentation	<input checked="" type="checkbox"/>	<input type="checkbox"/>
rtreportsadmin	<input checked="" type="checkbox"/>	<input type="checkbox"/>

At the bottom of the dialog are buttons for "OK", "Cancel", and "Help".

## 12.5 Knowledge check: Group and User Security

### Objectives

Given the local Windows Groups, configure the desired RtReports security and test their access.

### Problem Description

Five Domain Users were designated to perform specific actions in RtReports. They should only be allowed to do those activities and restricted from doing anything else. The users were already added to five different local groups in the RtReports Server (PIRTR01). The RtReports administrator must enable access permissions for each user as follows:

Windows Groups	Domain User	Permission
RtReportsApprovers	PISchool\LBeethoven	All permissions, except Admin
RtReportsCommenters	PISchool\LBernstein	User, Comments, and Print
RtReportsReviewers	PISchool\ACLBizet	User, Edit, Review, and Print
RtReportsUsers	PISchool\GFFVerdi	User and Print
RtReportsVerifiers	PISchool\WAMozart	User, Verifications, and Print

### Approach

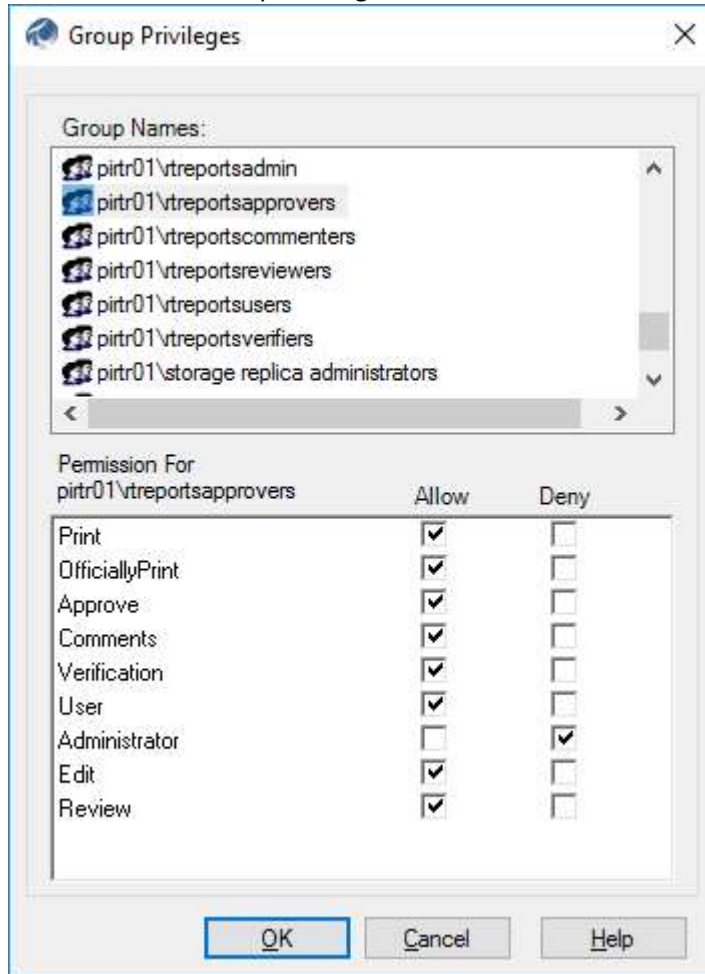
Use the RtReports Group Privileges tool to setup the access rights based on the table Windows groups.

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

**You can log into RtReports directly using their credentials. User Password: P1school! (for all users).**

**Step by step solution:****Setup Group Privileges in RtReports**

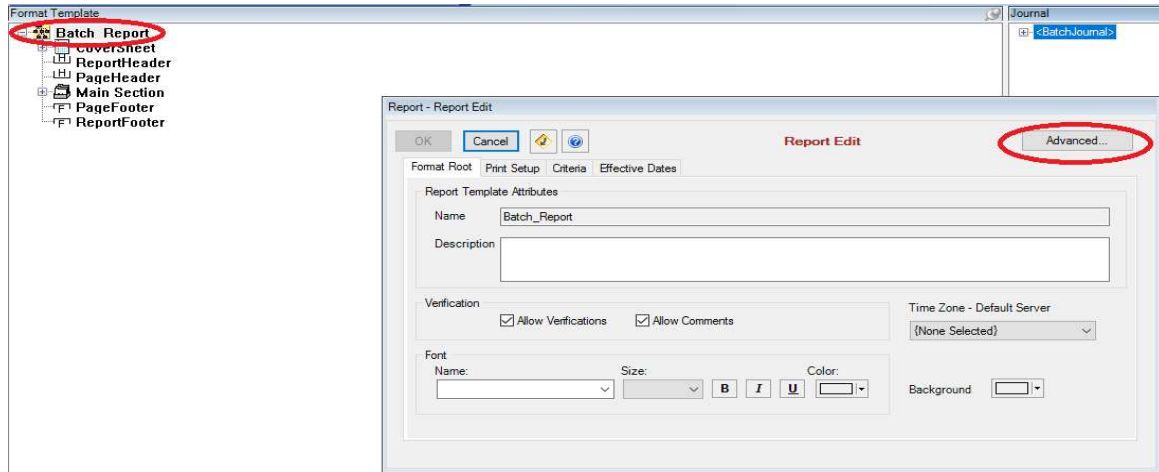
1. Login to RtReports Editor
2. Admin menu ► Group Privileges...



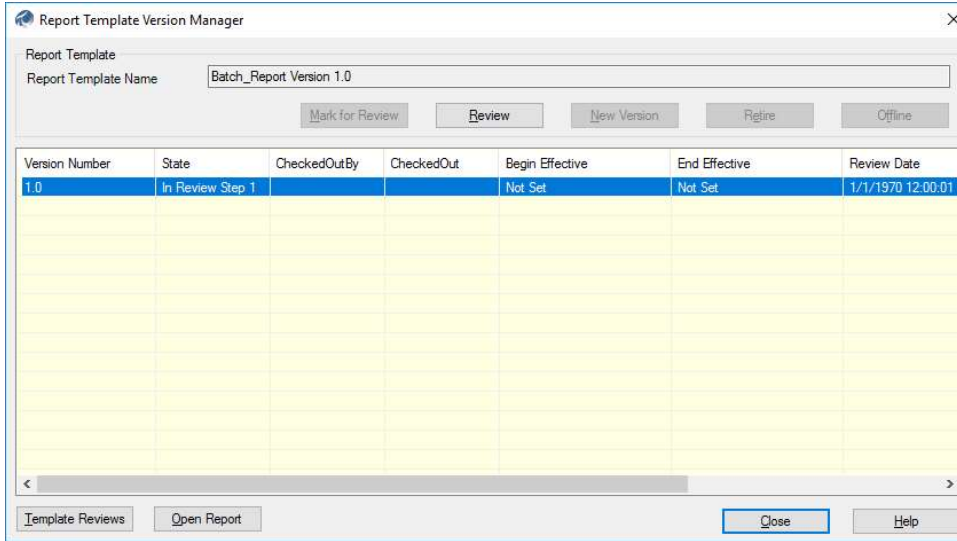
3. Under *Group Names*, browse the local group list ► select the RtReportsApprovers ► in the *Permission for <Domain>\<Group>* allow all permissions except the Administrator ► repeat:
  - RtReportsCommenters group and allow Comments, User, and Print permissions
  - RtReportsReviewers group and allow 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 Template Security

1. Launch RtReports Editor 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
4. Select Reviews tab ► click the + button ► browse and select RtReportsReviewers as the reviewing group
5. Select Approvals tab ► click the + button ► browse and select RtReportsApprovers as the approving group
6. Click OK ► OK

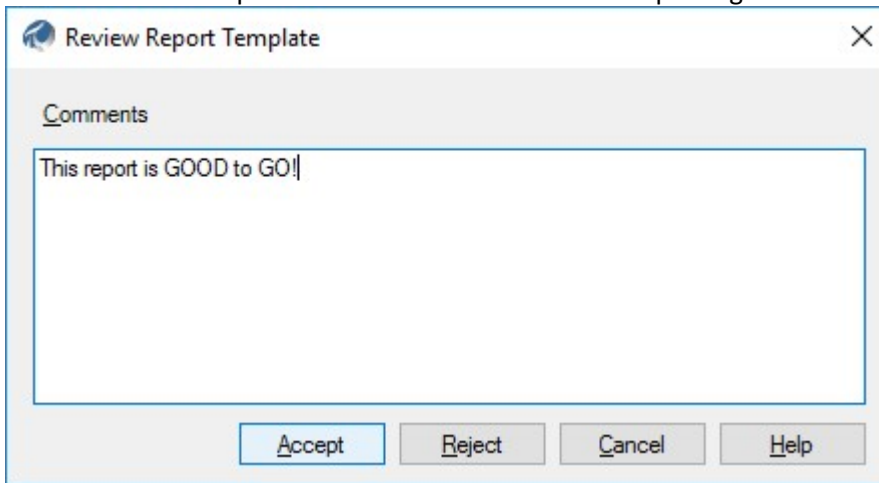


To begin reviewing the report: Tools ► Report Version Manager ► click Mark for Review ► Close ► Save the report. **Note:** the report is ready for Review and Release

7. Exit Report Editor

**Review and Release report**

1. Launch RtReports Editor 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

1. Launch RtReport Generator 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**)

The screenshot shows the 'Batch Search Page' in RtReports. On the left, there are search filters for 'Time Range' (State: Active, Start Time: \*-8h, End Time: \*) and 'Search Condition' (Data Server: PISRV01, Asset Server: PISRV01, Asset Database: THE Database, Product, Batch, Equipment, Operation, Phase). On the right, a table lists batches with columns: BatchID, Product, Recipe, StartTime, and EndTime. The row for 'BIDBDM0' is highlighted. The 'Reports' tab is selected in the top navigation bar.

BatchID	Product	Recipe	StartTime	EndTime
BIDRTS50	PrBDM50	RCSPI50	5/13/2019 9:14:50 AM	5/13/2019 11:01:00 AM
Pulp_BID-1140			5/13/2019 10:05:25 AM	5/13/2019 10:56:25 AM
BIDTLT55	PrDEK55	RCDEK55	5/13/2019 10:26:55 AM	5/13/2019 12:13:05 PM
CCF-BID: 71			5/13/2019 10:41:55 AM	5/13/2019 11:09:55 AM
Pulp_BID-2131			5/13/2019 10:58:25 AM	5/13/2019 12:03:25 PM
Pulp_BID-1141			5/13/2019 11:04:25 AM	5/13/2019 11:53:55 AM
BIDBDM0	PrTLT0	RCBDM0	5/13/2019 11:39:00 AM	5/13/2019 1:25:10 PM
CCF-BID: 73			5/13/2019 11:43:55 AM	5/13/2019 12:12:25 PM
Pulp_BID-1142			5/13/2019 12:03:25 PM	5/13/2019 12:55:25 PM
Pulp_BID-2132			5/13/2019 12:20:25 PM	5/13/2019 1:25:25 PM
CCF-BID: 75			5/13/2019 12:45:55 PM	5/13/2019 1:14:55 PM
BIDRTS5	PrSPI5	RCRTS5	5/13/2019 12:51:05 PM	5/13/2019 2:37:15 PM
Pulp_BID-1143			5/13/2019 1:02:25 PM	5/13/2019 1:48:55 PM

1. In the generated report ► click [Comment](#) hyperlink ► enter comments and signoff

The screenshot shows the 'Report Not Printed' page in RtReports. The top navigation bar includes 'Approve', 'Verifications', and 'Refresh' buttons. The main content area displays 'Main Batch Data' with a table:

Batch ID	Start Time	End Time
BIDTLT55	13-May-19 10:26:55 AM	13-May-19 12:13:05 PM

On the right side, there are hyperlinks for 'Comment' and 'Verify', and a 'Show Comment' checkbox. The 'No of Comment' is displayed as 0.

2. Notice that the [Verify](#) hyperlinks and [Approve](#) button are disabled (because LBernstein does not have those privileges)
3. Sign Off RtReports Generator

## Verify Report

1. Launch RtReports Generator 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"

The screenshot shows the RtReports web application interface. The top navigation bar includes 'View', 'Configure', 'Help', 'Support', 'Feedback', and 'Sign Off'. The user is logged in as 'student01'. The main content area is divided into a left sidebar and a main table.

**Time Range**

State: Active

Start Time: \*5h

End Time: \*

**Search Condition**

Data Server: PISRV01

Asset Server: PISRV01

Asset Database: THE Database

Product: [ ]

Batch: [ ]

Equipment: [ ]

Operation: [ ]

Phase: [ ]

**Report Condition**

**Batches Reports**

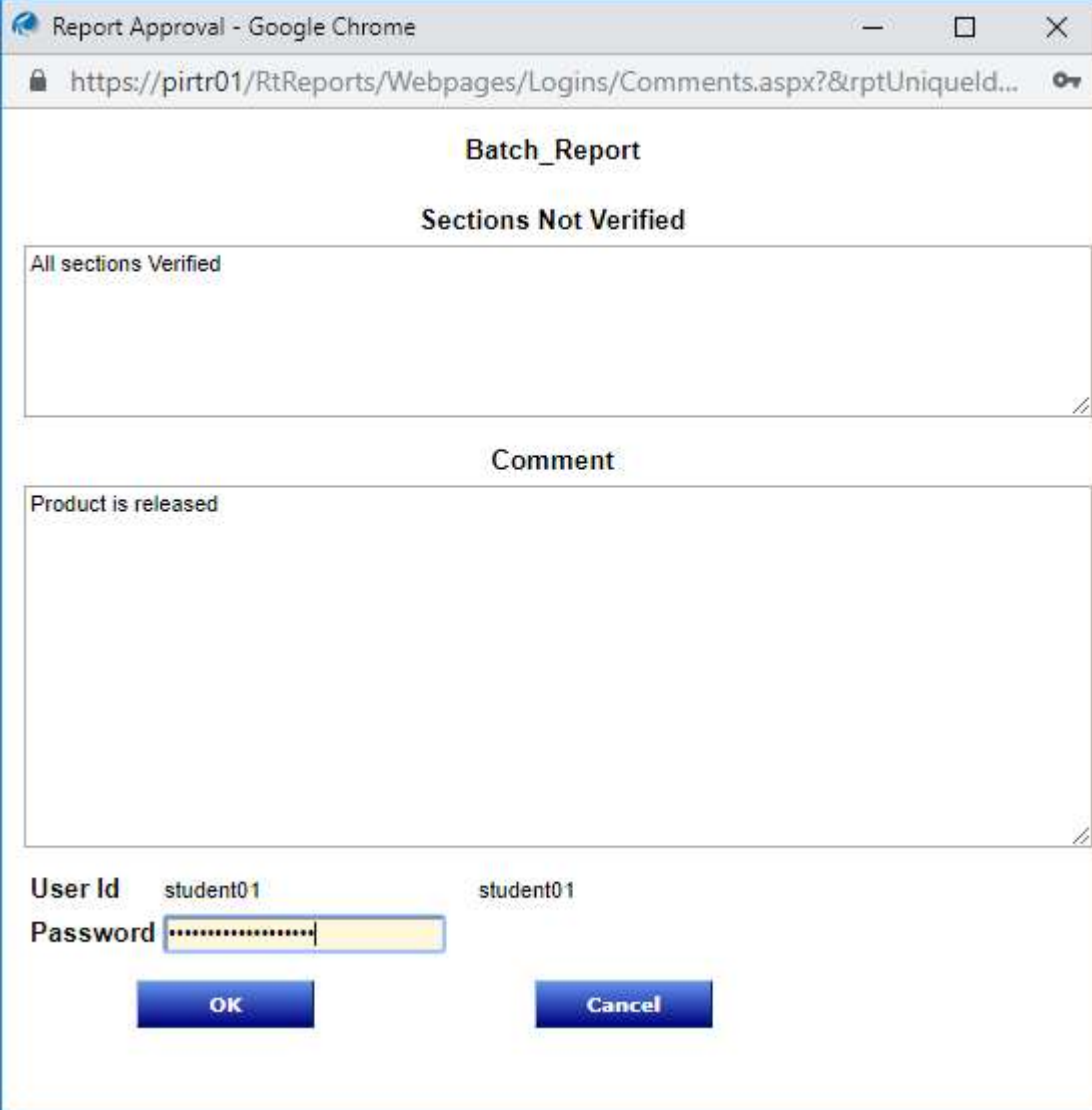
BatchID=BIDTLT55, Product=PrDEK55, Recipe=RCDEK55, Start Time=5/13/2019 10:26:55 AM, End Time=5/13/2019 12:13:05 PM

Report Name	Version	Description	Status	RunTime
<a href="#">Batch_Report</a>	1.0		Not Yet Approved	2.54
<a href="#">BatchRecord</a>	1.0		Not Run	3.62
<a href="#">Test_RNI</a>	1.0		In Edit	
<a href="#">testEF</a>	1.0		In Edit	
<a href="#">TestPageBreak</a>	1.0		In Edit	
<a href="#">TestSinWaveEF</a>	1.0		In Edit	
<a href="#">TQS</a>	1.0		In Edit	

3. In the generated report page, click the *Verify* hyperlink ► enter verifications...
4. Notice the Comment hyperlinks and Approve button are disabled (because WAMozart does not have those privileges)
5. Sign Off RtReports Generator

## Approve Report

1. Launch RtReport Generator 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



Report Approval - Google Chrome

https://pirtr01/RtReports/Webpages/Logins/Comments.aspx?&rptUniqueid...

**Batch\_Report**

**Sections Not Verified**

All sections Verified

**Comment**

Product is released

User Id student01 student01

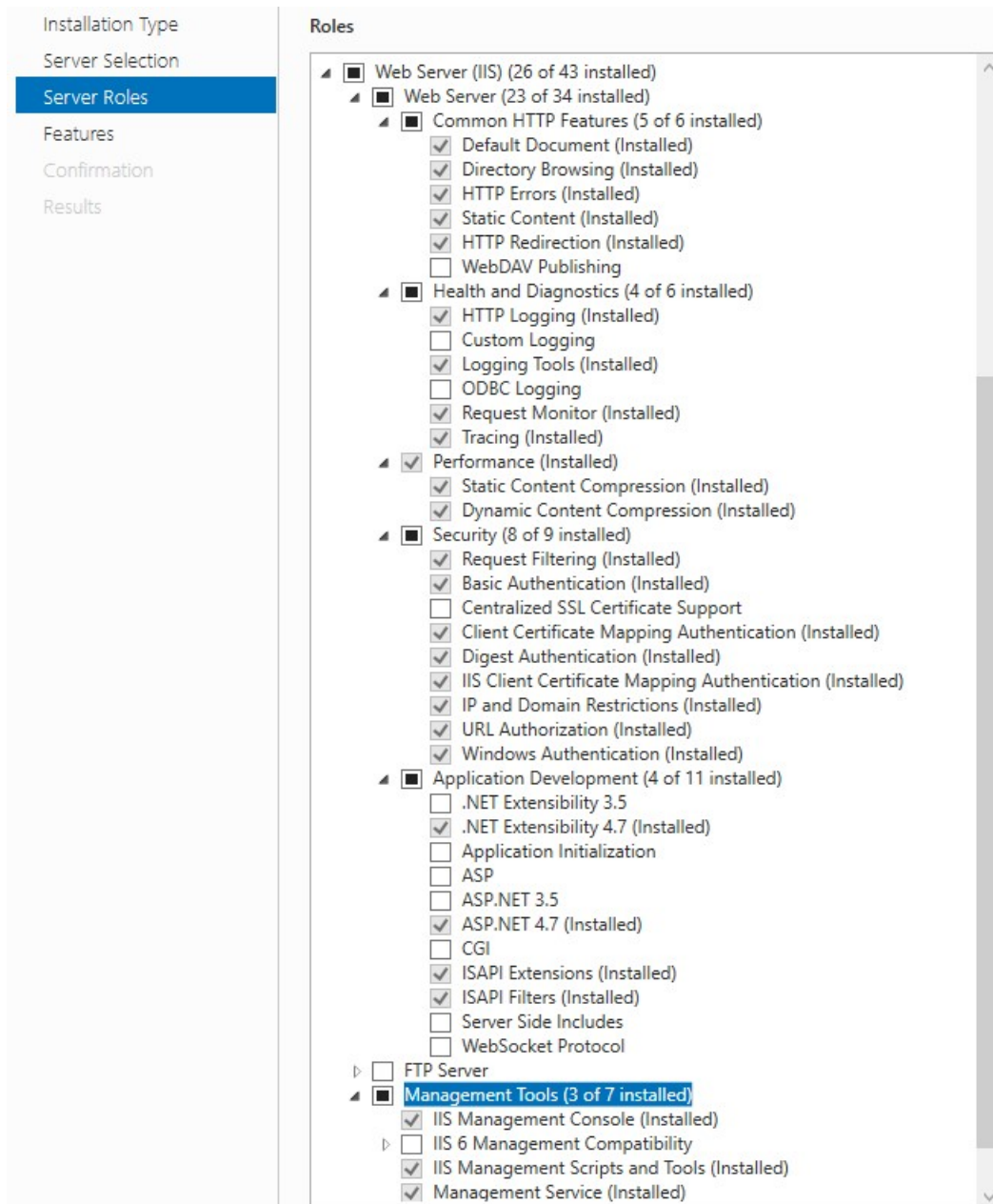
Password .....

OK Cancel

4. Notice the [Print](#) hyperlink is enabled; click the [Print](#) hyperlink. The .pdf **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

## 13. (Optional) RtReports Installation

- For detail, refer to documentation
- Login to the RtReports Server machine. **Note:** login user must have local Administrator privileges and PI Data Archive Admin privileges (or PI Trust with PI Admin privileges)
- Need a domain service account. **If using gMSA**, create a local service account <OSISoft.RtReports.RtRemotingHost> for the PI RtReports Remoting Host service



- Install Microsoft Internet Information Service (IIS)
- PowerShell x64 ► Run as administrator ► run **Set-ExecutionPolicy Unrestricted**

- PowerShell x86 ► Run as administrator ► run **Set-ExecutionPolicy Unrestricted**
- Run the RtReports Server Install Kit as Administrator. This will also install the *PI Baseline Services* ► prompts:
  - Welcome to RtReports x.x xxx – Server Setup: OK
  - Welcome to The ABCpdf .NET xx.x Setup Wizard: Next
  - License Agreement: check I Agree ► Next
  - Select Installation Folder: ... ► Next
  - Confirm Installation: Next
  - Installation Complete: Close
  - Welcome to RtReports x.x xxx Setup Wizard: Next
  - End-User License Agreement: ... ► check I accept the terms in the License Agreement ► Next
  - Compliance Acknowledgment: ... ► Next
  - Prerequisites Installation: ... ► check Install Windows Roles and Features (default) ► Next ► Install (will take sometimes) ► Windows Roles and Features installation has completed. ... ► OK
  - Destination Folder: ... ► Next
  - Service account: <domain\service account> and <password> ► Next
  - SSL Setup: SSL Port: 443 (default) ► Create (to create a Self-signed certificate) ► Yes ► OK (notice the created/selected Self-signed Certificate ► Next
  - PI Data Server for Module Database: PI Data Server: <PI Data Archive server> ► Next
  - Completed the RtReports x.x xxx Setup Wizard: ► Finish
  - Installation Complete: Close
- Open IIS Manager and configure identity for Application Pools: “PICR Print Trent”, “RtReports Admin App Pool”, and “RtReports App Pool”.
  - Be sure the service account(s) Read/Write access to %PIHOME%\RtReports folder
- Add RtReports Administrator member(s) to the local group RtReportsAdmin (created by the RtReports Installation)
  - Workaround, when using gMSA – change the PI RtReports Remoting Host Service’s Log on as to use the Local Service account
- Create 2 folders for Official Report Print and Unofficial Report Print, assign Read/Write privileges to the PI RtReports Remoting Host Service account and edit the values for **AdobePDFLocation** and **OfficialAdobePDFLocation** in the respective OSisoft.RtReports.RtRemotingHost.exe.config (found on RtReports\bin sub-folder)
- Optional for the file %PIHOME%\RtReports\web.config
  - **eMailErrorTo** – who to email to when RtReports encounters error

- Automatic Report Generation Settings:
  - **AdobePDFLocation** (matching the RtRemotingHost.config file)
  - **ARGSenderEmail** – Auto Report Generation Email Sender
  - **Network host** – SMTP Mail Server FQDN, Hostname, or IP...
- Edit file C:\Windows\System32\inertsrv\config\applicationHost.config:
  - Under <**system.applicationHost**>, set serviceAutoStartEnabled = "true"
- Restart IIS

### 13.1 Background: RtReports Server Settings

Few essential RtReports Server settings (defined in %PIHOME%\RtReports\web.config). Refer to RtReports Documentation for more.

Parameters	Information
AdobePDFLocation	<b>Folder where to print/store the Unofficial report .pdf files</b>
OfficialAdobePDFLocation	<b>Folder where to print/store the Official report .pdf files</b>
AllowCommentandVerify	<b>Default: true</b>
AllowCommentAfterApprove	<b>Default: false</b>
AllowMultipleVerification	<b>Default: true</b>
AllSectionVerifiedToApprove	<b>Default: false</b>
ARGSenderEmail	The ARG notifications email Sender Email Address
AllowHTMLMarkup	Allows Users to insert HTML formatting into the display of the RtReports Editor. <b>Default: true</b>
DefaultStartTime	The default query Start Time used in the Generator. <b>Default: *-8h</b>
eMailErrorTo	Send the RtReports System Error To email address
ImageFileSettings	Allow to add an image on the logon page
InitialBatchSearch	RtReports Generator runs Batch Search on opening the Batch Search page. Default: true
InitialPage	At login, open Batch or Report search page.

Parameters	Information
	<b>Default: Batch</b>
Network Host	The SMTP email Server for relaying ARG emails
PreferredSearchMode	At login, if InitialPage=Batch and InitialBatchSearch=true, runs Batch or EF search. <b>Default: Batch</b>
ReportTimeZone	Report using <b>Server</b> or <b>Client</b> Time zone <b>Default: Server</b>
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
UserAuthenticationRoot	Computer used to lookup group security attributes

## 13.2 Background: RtReports Cache Management Settings

Report Cache Management (defined in %PIHOME%\RtReports\Web.config)

Parameters	Information
StorageSize	Number of reports cached before they are purged. <b>Default: 200</b>
Directory	Location of the cached reports <b>Default: %PIHOME%\RtReports\cache</b>
ReloadCache	Controls whether the existing cached reports are reloaded from the files when the RtReportRunner and CacheManager are restarted <b>Default: true</b>

## 13.3 Background: RtReports Editor Standalone Installation

- For details, please refer to documentation
- Login to the user computer (client)
- Run the RtReports Editor Install Kit (Run as administrator)
- Follow prompts ► Web Server: <RtReports Server Hostname> ► PI Data Server: <The PI Data Archive Hostname> ► ... ► Installation complete

## 14. **Final Exam**

The final exam in this course is taken online. Please check the course listing online for more details.