



PI World 2020 Lab

See the Light, Easy wins to improve your Asset Framework Experience

OSIsoft, LLC 1600 Alvarado Street San Leandro, CA 94577

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

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

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

U.S. GOVERNMENT RIGHTS

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

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

Table of Contents

Ta	able of	Conte	ents	3
1.	Intr	roduct	ion	5
	1.1.	Ove	rview of this Lab	5
	1.2.	Why	adopt Asset Framework for your enterprise?	5
	1.3.	Poss	ible modelling approaches	6
	1.3.	.1.	Bottom-up approach	7
	1.3.	.2.	Top-down approach	7
	1.3.	.3.	Value-oriented approach	8
2.	Bui	lding y	our Asset Framework database	9
	2.1.	Why	AF element templates	9
	2.2.	Our	case study – a sugar and salt packaging company	. 11
	2.3.	Asse	t monitoring in PI Vision - Directed activities	. 14
	2.4.	Asse	t Framework Best Practices - Directed activities	. 24
	2.4.	.1.	AF templates	. 24
	2.4. 2.4.	.1. .2.	AF templates More on handling variations – Allow Extensions vs Derived Templates	. 24 . 35
	2.4. 2.4. 2.4.	.1. .2. .3.	AF templates More on handling variations – Allow Extensions vs Derived Templates Units of Measure	. 24 . 35 . 42
	2.4. 2.4. 2.4. 2.4.	1. 2. 3. 4.	AF templates More on handling variations – Allow Extensions vs Derived Templates Units of Measure AF Tables	. 24 . 35 . 42 . 47
	2.4. 2.4. 2.4. 2.4. 2.4.	1. 2. 3. 4. 5.	AF templates More on handling variations – Allow Extensions vs Derived Templates Units of Measure AF Tables Enumeration Sets	. 24 . 35 . 42 . 47 . 57
	2.4. 2.4. 2.4. 2.4. 2.4. 2.4.	1. 2. 3. 4. 5. 6.	AF templates More on handling variations – Allow Extensions vs Derived Templates Units of Measure AF Tables Enumeration Sets NERD ALERT - String Builder	. 24 . 35 . 42 . 47 . 57 . 61
	 2.4. 2.4. 2.4. 2.4. 2.4. 2.4. 2.4. 2.4. 2.4. 	1. 2. 3. 4. 5. 6. 7.	AF templates More on handling variations – Allow Extensions vs Derived Templates Units of Measure AF Tables Enumeration Sets NERD ALERT - String Builder OPTIONAL - Categories	. 24 . 35 . 42 . 47 . 57 . 61 . 65
3.	2.4. 2.4. 2.4. 2.4. 2.4. 2.4. 2.4. 2.4.	 1. 2. 3. 4. 5. 6. 7. orking s 	AF templates More on handling variations – Allow Extensions vs Derived Templates Units of Measure AF Tables Enumeration Sets NERD ALERT - String Builder OPTIONAL - Categories	. 24 . 35 . 42 . 47 . 57 . 61 . 65 . 70
3.	2.4. 2.4. 2.4. 2.4. 2.4. 2.4. 2.4. Wo 3.1.	1. 2. 3. 4. 5. 6. 7. 7. why	AF templates More on handling variations – Allow Extensions vs Derived Templates Units of Measure AF Tables Fnumeration Sets NERD ALERT - String Builder OPTIONAL - Categories with Asset Analytics	. 24 . 35 . 42 . 57 . 61 . 65 . 70 . 70
3.	2.4. 2.4. 2.4. 2.4. 2.4. 2.4. 2.4. Wo 3.1. 3.2.	1. 2. 3. 4. 5. 6. 7. 9rking 9 Why Asse	AF templates More on handling variations – Allow Extensions vs Derived Templates Units of Measure AF Tables Enumeration Sets NERD ALERT - String Builder OPTIONAL - Categories with Asset Analytics Asset Analytics	. 24 . 35 . 42 . 57 . 61 . 65 . 70 . 70 . 71
3.	2.4. 2.4. 2.4. 2.4. 2.4. 2.4. 2.4. Wo 3.1. 3.2. 3.2.	1. 2. 3. 4. 5. 6. 7. 7. why Asse 1.	AF templates More on handling variations – Allow Extensions vs Derived Templates Units of Measure AF Tables Enumeration Sets NERD ALERT - String Builder OPTIONAL - Categories with Asset Analytics Asset Analytics t Analytics Best Practices - Directed activities Use of Expression Variables	. 24 . 35 . 42 . 47 . 57 . 61 . 70 . 70 . 71 . 71

Save the Dat	tel Fror	Bookmark not defined
Additiona	l resources	
References .		94
Goodbye!		93
The work yo	u have done	
3.2.5.	NERD ALERT - Analyses in Error	
3.2.4.	NERD ALERT – Analysis Organization and Consolidation	
3.2.3.	Rollup Analyses	

1. Introduction

1.1. Overview of this Lab

This Lab offers you a wealth of best practices that will substantially improve your experience of the PI System. We will focus on **Asset Framework** (a.k.a. **AF**) and **Asset Analytics**, and we will use PI Vision to look at the consequences of adopting good and bad practices in AF modelling. We will work on AF using **PI System Explorer** (a.k.a. **AF Client**).

The Lab is conceived for users that already have some hands-on experience with Asset Framework and Asset Analytics. Should you feel insecure on some aspects of this Lab, please consult the References and Additional resources sections at the end of this workbook.

Some sections are marked by a **NERD ALERT**! For time constraints, during the Lab we won't cover the topics described in these sections. If you are already familiar with the concepts illustrated in some of the exercises, feel free to go through the NERD ALERT sections yourself.

You should be able to reproduce each step of this Lab by following this workbook. Should you get stuck or want additional clarifications, please do not hesitate to ask questions to the Lab instructors and helpers.

1.2. Why adopt Asset Framework for your enterprise?

The introduction of Asset Framework in 2008 has marked the passage from offering a "simple" data historian (the Data Archive) to providing a **full data infrastructure**, which can be used at all levels of an organization to access data in an intuitive, user-friendly way.

A data infrastructure extends the scope of your PI System way beyond site-level data historians, by (1):

- creating a universal, master data system. An infrastructure simplifies governance so important data can be formally managed. It also supports processes that democratize data so people in any organizational department or at any level have real-time access to data and information
- removing the work associated with finding, converting and organizing data. People no longer have to wrestle with complex data access chains to find the "right" data for analysis or reporting
- making data available for multiple purposes. Instead of being used for preordained purposes, data can be accessed and shared with people and systems to support systematic enterprise control and communication throughout its parts

- reducing overall complexity and cost. An infrastructure eliminates the skilled resources, customized solutions and coding associated with connecting data from isolated point solutions, applications and historians
- creating enterprise readiness. Enterprises can rapidly take advantage of advances in IT applications, technology and solutions without having to re-integrate or rip and replace enterprise operational technologies (OT) architectures.

With its 10+ years of history, Asset Framework has become a crucial component of the PI System for its capacity of structuring operational data from the Data Archive and integrate this information with contextual metadata.

In combination with other components of the PI System (and specifically Asset Analytics and Notifications), Asset Framework is being used by our customers for a variety of purposes, such as:

- condition-based maintenance
- asset-health real-time monitoring and anomaly detection
- performance optimization
- diagnosis of downtime events
- data modelling and data cleansing for integration with third-party advanced analytics

It is important to mention that the successful deployment of AF across the enterprise depends on an initial investment in brainpower. Asset Framework is not a turn-key solution because it aims at modelling any industrial process, for a large array of business needs. When you start modelling in AF, you needa strategy to make sure that you quickly derive business value from it. What are the possible approaches?

1.3. Possible modelling approaches

If you've worked with the PI System before, most likely you are already familiar with what a Data Archive and a PI Point are. If not, here is a brief definition of both:

Data Archive: the component of the PI Server that stores and archives time-series data and serves it in real time throughout the PI System and your information infrastructure.

PI Point (a.k.a. **PI Tag**): the **basic building block of a PI System**. A PI Point represents a single data stream coming from a data source and being tracked by the PI System. Examples of PI Points could be the temperature of a bearing ring, the state of a digital alarm or the result of a KPI calculation.

1.3.1. Bottom-up approach

When a company's data management team (meaning, the PI administrators and possibly a few *power users*) decides to adopt and deploy Asset Framework, normally it's been using Data Archive for a while and it has created a certain amount of PI Points. Since the PI Points already exist, one approach to AF modelling would be to group all PI Points by asset, and then attempt to group the assets by similar PI Points. These "similar-object groups" become AF element templates (more about AF templatesin Section 2.1), and the "objects" become AF elements.

Typically, the following step is to layer static metadata (such as the asset type, its installation date, the date of last maintenance and so on), KPI calculations and other calculations needed for real-time notifications.

This **bottom-up approach** has the advantage of being a relatively clear, reality-based path to follow.

We know the PI Points, and we know we can layer analyses on top of them... but where's the vision? Why are we doing any of this? And what implies that the result we happen to create will be at all useful?

When we start with what we have instead of what we want – we don't know what we are missing (what other data sources could be valuable?), and we come to the problem with a **mindset dominated by the instrumentation**, which may not be appropriate for process-decision support.

Also, we might quickly abandon the task as we get overwhelmed by the sheer number of PI points that we want to group by similar assets.

1.3.2. Top-down approach

Alternatively, some users start modelling AF with a focus on the enterprise hierarchy – they put the corporate name at the top, then build a logical tree and at the bottom of the tree they attach the individual equipment units. The good news is that this **top-down approach** focuses on a goal – building a data infrastructure.

The less than good news is that the requirement – the expectation of value is still not that well defined. For some customers **this process of sorting into the hierarchy can take a long time** and, in the meantime, the "incomplete" hierarchy is not that useful, even as a data infrastructure. This approach may postpone the business value until the data infrastructure is "complete"; this can result in a long wait with delayed value, cutting corners to meet deadlines, etc. Also, creating a very ramified hierarchy in AF, with e.g. 8 levels to drill down to get to the equipment unit of interest, would result in a lot of clicks and a frustrating end-userexperience (we'll come back to this while working in PI Vision).

1.3.3. Value-oriented approach

The approach we recommend is the **value-oriented approach**. Whatever work you were doing **before**you started to use Asset Framework – we encourage you to keep working with the same goal in mind and not expect to postpone the business outcome for very long.

For example, we could build an analysis directly into Asset Framework (using Asset Analytics) instead of creating it in many different Excel spreadsheets using Excel and PI DataLink functions. This will help you create a centralized, consistent version of the truth on the AF Server, while PI DataLink will be mainly used to present the final results (i.e. mainly as a visualization tool, and not as a calculation tool).

Also, we can templatize AF elements and analyses with the goal of building reusable PI Vision and PI ProcessBook displays rather than multiple, tag-based displays.

The idea here is to initially build our AF templates with a simple, clear project in mind, and expand the scope of our Asset Framework project as we knead new requests into the mixture and identify new value opportunities.

Ultimately – implementing Asset Framework is not so much a project as a paradigm shift. It's not so much "bottom up" or "top down" but how to reconcile end-user efforts with internal company standards. It's reconciling the ultimate data infrastructure with the needs of value-giving analyses and display projects; reconciling the strategic with the tactical.

2. Building your Asset Framework database

This Chapter tackles many of the most common questions that you might encounter when modelling your AF Server. Section 2.1 highlights the importance of using AF templates, Section 2.2 presents the use case created for this Lab, Section 2.3 guides you in navigating PI Vision displays in order to identify a few issues that we solve step by step in Section 2.4. Each exercise focuses on a specific aspect of AF administration.

2.1. Why AF element templates

How long does it take to deploy a new asset and integrate it with existing displays?

How easily can you tell whether the problem is due to a single bad actor, or if there is a larger pervasive issue with a certain equipment manufacturer or region?

The ability to reliably bring new units online or run asset-to-asset comparisons relies on standardization. These standards should be simple to implement, deploy, govern, and update from a central location. With Asset Framework, you can easily create a view of your asset with associated metadata that integrates and contextualizes data from multiple repositories, whether from the Data Archive or other business or maintenance systems. These asset associations can then be turned into a template that applies to other similar assets, so that every boiler, every pump, or every transformer has a reliable, consistent view. Leveraging templates means that when youcreate a new efficiencycalculation, that same calculation can be applied automatically, consistently to all units with the same template . If new sensors or a different maintenance database is added, these updates can be managed centrally, thereby reducing administration time and errors. (2)

AF templates should be used even when the template has no attributes in the beginning. For example, you could initially build a template for all the elements representing a site. Further on in the construction of your AF database, you might decide to perform some rollup calculations at the site level (see for example the exercise in Section 3.2.3). Having a template ready for this will make maintenance easier.

For whom is already familiar with other PI software components, here are some practical advantages when using AF templates in the PI System:

- build **templatized analytics** (<u>YouTube tutoria</u>l).
- reuse PI Vision displays for similar assets (LiveLibrary and YouTube tutorial)
- use Collections in PI Vision displays (LiveLibrary and YouTube tutorial)
- use smart hyperlinks in PI Vision displays (LiveLibrary and YouTube tutorial)

- build PI ProcessBook Element Relative Displays (LiveLibrary and YouTube tutorial)
- reuse PI DataLink reports for similar assets (LiveLibrary and YouTube tutorial)
- use the **PI Integrator for Business Analytics** to filter, cleanse and publish data to external databases, advanced analytics platforms and data lakes (LiveLibrary and YouTube tutorial)
- use more efficiently PI Developer Technologies such as PI AF SDK, PI Web API, PI SQL Client and PI OLEDB Enterprise (general overview <u>here</u>, <u>Hands-on Lab presentation</u> and <u>PI SQL Online</u> <u>Course</u>)

Last but not least, using AF templates greatly improves the **overall performance of the AF Server**. Using templates reduces data duplication in the backend SQL database and drastically improves the performance of searches in client applications.

If you are not familiar with some of the terms or products above, feel free to click on the linked pages or to speak with the instructors and other OSIsoft employees at the end of the Lab. 2.2. Our case study – a sugar and salt packaging company

The packaging company Sweet & Savory Corp has two core products: bags of sugar and bags of salt.

The company uses FORM, FILL and SEAL (FFS from now on) equipment units to produce the bags.



Figure 1. Form, Fill and Seal (FFS) unit (source: (3))

In principle, an FFS machine is a relatively simple unit and it is extensively used in packaging lines.

The material that is to be packaged (in our case salt and sugar) is loaded into a hopper above the machine. A controlled amount is let down into a loading cartridge situated just above the package opening.

The packaging material (normally plastic) is loaded into the machine as a roll and **formed** into a pocket by a series of rollers. The loading cartridge will open above the package opening and deposit the material into a pocket, **filling** it.

A heated sealer will then clamp the plastic above the pocket, **sealing** the package just before a knife cuts the pocket free to drop onto a conveyor where it is taken off for palletizing. This process is then repeated until either the hopper is empty, or the machine runs out of packaging material.

If you'd like to see an FFS unit in action, feel free to watch this YouTube video: <u>1KG flour Vertical FFS bag</u> packing machine automatic with auger filler granule filling packaging.

The FFS units used by Sweet & Savory Corp come in two models: ADCO and MF TECNO. The models present similar technical specifications but are manufactured by two different suppliers.

The **chief maintenance engineer** of Sweet & Savory Corp is working on a large-scale condition-based maintenance program and needs to standardize the modelling and monitoring of the FFS units.

The project will be initially deployed in one production site, and if the outcome is positive, it will be deployed at an enterprise-scale level across the globe.

The units should be organized by:

- Supplier (in order to facilitate the filing of work orders)
- Type of product (in order to monitor production and fine-tune the machines' settings)

Also, the chief maintenance engineer would like to integrate the PI System with maintenance information coming from an external table hosted in a SQL Server. The information contained in the external table is reported in Table 1.

Table 1. Maintenance information for each FFS unit

Asset	Model	Serial Number	Installation Date	Last Service Date	Service Crew
FFS01	ADCO	ADA5FFS01	2017-05-01	2019-11-01	Red

For his purposes, the chief maintenance engineer would like to retrieve the **Last Service Date** for the site in question.

The same AF database is also accessed by the **chief process engineer**, who wants to monitor:

- the number of bags packed by each FFS unit per minute
- the **number of bags** packed by each FFS unit **since the beginning of the day**
- the **moisture content** (which should be kept below 0.1%)
- the grain size. This is required only for the sugar bags
- the **bags size** (which can be 220, 500 or 1000 g)
- the Overall Equipment Effectiveness (OEE) of the machines. The Overall Equipment
 Effectiveness (OEE) is calculated as: OEE = Availability x Performance x Quality

In addition, he would like to keep track of the **operator** working in the production line on every shift for semesterly performance evaluation.

The data management team of Sweet & Savory Corp has already been working on the company's AF database, and at the moment around 20 FFS units are modelled are monitored.

The team is actively working in a **sandbox environment** to test some calculations and configuration aspects of six FFS units (FFS01 to FFS06).

For simplicity and ease of navigation during the Lab, the sandbox environment is here modelled as a branch of the production AF database. In the real world, the sandbox environment would reside on a dedicated AF database, often hosted on a dedicated AF server.

The exercises presented in this Chapter will start and end with the end-user experience in PI Vision, our web-basedvisualization tool. PI Vision will give us clear indications on whether something has gone wrong in the AF modelling. Specifically, **the sandbox environment is currently affected by a few issues**, which we want to first identify and then address in Section 2.4.

2.3. Asset monitoring in PI Vision - Directed activities

Step 1. Open PI Vision from the shortcut link on the desktop of the virtual machine. The first time you open the page it might take a while!



Step 2. Once on the main page, open the display Sandbox_Overview_FFS (Figure 2).

PI Vision	
Show private displays	All Displays (4)
Search All Displays	
Filter by Keywords	Form Fill & Bear units - production overview
🖽 All Displays	
☆ Favorites	
오 My Displays	
() Recent	
🖬 O 🗑	Sandbox_Overview_FFSSandboxPISCHOOL\student01PISCHOOD
☆ Home	🔔 🌾 🗘 🔶
- 📷 Imported ProcessBook Displays	
PI World SF 2020	

Figure 2. PI Vision main page

This display (Figure 3) provides an overview of the current production from all the FFS units. We can verify which units are underperforming given the daily packaging target, and which units are reporting a low OEE (Overall Equipment Effectiveness).



Figure 3. Sandbox_Overview_FFS display

Step 3. Click anywhere in the tile showing FFS01:



This will open a more detailed display, Sandbox_FFS_Monitoring (Figure 4).

The display reports the trends of the main process variables associated with FFS01, such as the Bags Per Minute, the Sealer Temperature, the Moisture Content etc. The display also offers contextual metadata such as the model, the last service date, the bag size etc.



Figure 4. Sandbox_FFS_Monitoring display

Instead of creating individual displays for each and every asset, PI Vision allows you to reuse a display for all assets presenting similar characteristics, or in PI jargon: *"all AF elements using the same AF template"*. Use the **Asset Dropdown Menu** to switch asset and see the display update with the relevant information.

Step 4. Choose **FFS02** and verify that the trends update to reflect the chosen asset.

We know that the sandbox environment includes FFS units from 01 to 06.

Step 5. Try switching to FFS05 using again the Asset Dropdown Menu. Is this asset available?

We identified the first issue to tackle!

Issue #1	. Asset	FFS05	is not	available	e in the	PI Visi	on Asset	Dropdown	Menu.
----------	---------	-------	--------	-----------	----------	---------	----------	----------	-------

Sandbox_FFS_Monitoring Asset:	FFS01 ▼
	Switch Asset
	From
	FFS01
	То
	FFS02
	FFS03
ien a hud all article	FFS04
-50	FFS06

Let's further explore our displays to look for additional issues.

Step 6. Go back to the display Sandbox_Overview_FFS using the "back to overview" button.

		ŪŪ	
Packaging	back to overview		

Step 7. Look at the table on the bottom-right corner, which gives us the Sugar Grain Size statistics for

the sugar packaging units in the sandbox environment (FFS03 to FFS06).

Step 8. In order to better inspect the Trend column, set the time span to 1 hour.



Do you notice anything strange? We've found the 2nd issue to solve!

Issue # 2. Trends and statistics for FFS units 3,4 and 6 are identical.

Sugar FFS Units							
Name 🔺	Trend	Average	Minimum	Maximum	StdDev	Units	
FFS03 Sugar Grain Size		0.49143	0	0.68254	0.29128		
FFS04 Sugar Grain Size		0.49143	0	0.68254	0.29128		
FFS05 Sugar Grain Size	J-J	0.54208	0	0.69801	0.26178	mm	
FFS06 Sugar Grain Size		0.49143	0	0.68254	0.29128		

Let's proceed in our diagnostics.

Step 9. Click on the tile for FFS01 to go back to the **Sandbox_FFS_Monitoring** display.



Look at the trend of the **Sealer temperature**. This temperature is expected to vary **between 70 and 270** °F. Is it operating within the expected range? Auch, one more issue!



Issue # 3. The attribute Sealer temperature shows values outside the expected range.

The chief maintenance engineer would like to include the Last Service Date attribute in the Table symbol.

Name	Value	Units
FFS01 Model	ADCO	
FFS01 Shift	Shift 2	
FFS01 Operator	Toni	
FFS01 Nominal Max Throughput	75	Bags/min
FFS01 Packing Rate Per Minute	55	Bags/min

In order to add the Last Service Date attribute to the table (see also Figure 5):

- 1) Switch to Edit mode (button in the top-right corner of the display).
- 2) Open the asset pane on the left side of the display.
- 3) Navigate to Sweet & Savory Corp \rightarrow Sandbox \rightarrow Packaging \rightarrow FFS01.

(use the little arrows to drill down to the next level a = count is counter by contract the finance <math>(1))

🖌 藚 Sweet & Savory Corp

- 4) Select the Last Service Date attribute from the Attributes pane (bottom left pane)
- 5) Drag and drop it on top of the Table symbol (you should see it displayed as

Last Service Date

NB: Do this operation only once! If the display doesn't update, please give it some more time.



Figure 5. How to add an attribute to an existing symbol in PI Vision

Did you experience any delays when adding the attribute to the display? This brings us to the 3rd issue:

Issue # 4. The attribute Last Service Date slows down the loading of the PI Vision display.

Some gory details below...

NERD ALERT – Data Caching in PI Vision

The performance issue above normally affects the initial loading of the display; the delay could seem insignificant when refreshing the page. This is due to data caching, a mechanism that helps reducing network workload by keeping a copy of frequently used data on the client machine.

The loading time of a PI Vision display depends both on the caching of data on the PI Vision server itself, and on the caching of data on the web browser of the client machine. You can check the "fresh" loading time of our PI Vision display by clearing the web browser cache (*Ctrl* + *Shift* + *Del* \rightarrow Clear Data) and the PI Vision cache (e.g. by recycling the Application Pools or running an *iisreset* from an elevated command prompt). You should see that the Sandbox_FFS_Monitoring display takes a long time to fully load.

dbox	FFS_Monitoring						
1							
						back to	overview
2							
г							
							Packing Rate Per Min
							~
							Moisture Content
	Name	Value	Units				
4				\bigcirc	E	\rightarrow	\bigcirc
, i					(())	()
					(・	
ſ				\diamond	\diamond	\checkmark	$\otimes $
				FFS02 Bags Today	FFS02 Through	nput Efficiency	FFS02 OEE

- Step 10. Save the change to the display by clicking on the of the display.
- Step 11. Switch to unit FFS04 (using the Asset Dropdown Menu and look at the table in the bottom-left corner. Do you notice any problems?

Issue # 5. Asset FFS04 shows "No Data" for the Nominal Max Throughput.



button on the top-right corner



Name	Value	Units
FFS04 Model	MF TECHNO	
FFS04 Shift	Shift 2	
FFS04 Operator	Toni	
FFS04 Nominal Max Throughput	No Data	Bags/min
FFS04 Packing Rate Per Minute	53	Bags/min
FFS04 Last Service Date	11/1/2019 12:00:00 AM	

Let's move on to the next issue and don't worry... we will tackle these problems one by one in the coming sections!

Recently the chief maintenance engineer tried to build a new display, but he encountered difficulties with the display titles.

Step 12. Click on



and open the PI Vision display

Sandbox_FFS_Monitoring_Faulty_Title (the displays are organized from left to right in order of last access, so this display could be on the rightmost of the main page if you haven't opened it already).

PI Vision			
Show private displays	All Displays (4)		
Search All Displays			
Filter by Keywords	tak Kontons	Form Fit & Said units - production overview	FFS04 Sugar Packaging bestautone
🖽 All Displays			
ත් Favorites			
A My Displays			
() Recent			
■ 0 m	Sandbox_FFS_Monitoring PISCHOOL\student01	Sandbox_Overview_FFS PISCHOOL\student01	Sandbox_FFS_Monitoring_FaultyTi PISCHOOL\student01
☆ Home	👗 🕷 🗘 🔶	👗 🕷 🔅 🔶	👗 🖗 🗘 🔶
- m Imported ProcessBook Displays			

Step 13. Switch between FFS units using the Asset Dropdown Menu.

What happens to the title of the display?

Issue # 6. The title of the display Sandbox_FFS_Monitoring_Faulty_Title does not update with the chosen asset.



Now that we have a few issues to solve, let's move to Asset Framework and sort them all out!

- Asset Framework Best Practices Directed activities 2.4.
- 2.4.1. AF templates

The first issue that we want to address is:

FFS04 FFS05

🗊 FFS06

Issue # 1. Asset FFS05 is not available in the PI Vision Asset Dropdown Menu



It might take some time for the Attributes tab to get populated with values (Figure 6). We will come back to this issue later on in the Lab (Section 2.4.3).

🗉 Asset

📃 Last Service Da

FFS01								
Gene	eral Ch	ild Elements	Attributes	Ports	Analyses	Notification Rules	Version	
Filter	r							
	/: =	🔶 💂 Name		⇔ Va	ue			
	🖻 Ca	itegory: Iden	tification					
	۰ چې	A	sset	Re	trieving the d	ata reference value	s. Please wait	
	٠,	🔳 La	ast Service D.	. Re	trieving the d	ata reference value	s. Please wait	
		I M	odel	AD	со			
	۰,		ominal Max T.	Re	Retrieving the data reference values. Please wait			
	ک ې 🖬	🔳 Pr	oduct	Re	trieving the d	ata reference value	s. Please wait	
	ک ې 🖬	📃 Pr	oduction Line	Re	trieving the d	ata reference value	s. Please wait	
	🖻 Ca	tegory: Over	rall Equipment	Effec	tiveness			
Ð	۰,	🔶 🍼 A	vailability	Re	trieving the d	ata reference value	s. Please wait	
⊞	۰,	🔶 🍼 A	vailability Mo.	. Re	trieving the d	ata reference value	s. Please wait	
	۰,	🔶 🍼 o	EE	Re	trieving the d	ata reference value	s. Please wait	



Step 4. Switch to FFS05 to compare the attribute list. What do you notice? (see also Table 2)

Step 5. Click on the General tab and compare the two elements. (see also Table 2)

FFS05				
General	child Elements	Attributes	Ports	A
Name:	FFS05			
Description	:			
Template:				

Table 2. Comparison between FFS01 and FFS05



You probably observed that the two elements do not show exactly the same attributes and do not use the

same attribute naming convention. In FFS01 you'll notice a small symbol next to each attribute, which means that the attribute is based on an AF element template. FFS01 is based on the AF element template **Sandbox_FFS** (visible under the General tab), while FFS05 does not use any template.

This discrepancy is due to a workaround adopted in the modelling phase: FFS05 has some issues with the PLC, and not all readings are available. In particular, the moisture content is not available. For this reason, the data management team decided to build FFS05 from scratch (without using the AF element template) and created only the attributes for which measurements are available.

The problem is that, over a few months' time, the structure and nomenclature of the AF element template was modified compared to the manually created element FFS05.

Creating AF elements without using an AF element template presents several drawbacks. For example, FFS05 will not be recognized by PI Vision as an FFS unit, and any change made to the FFS AF element template will not propagate to FFS05.

A better solution is to (a) assign FFS05 to the same AF element template as its siblings and (b) exclude the attributes that are not available for FFS05.

Step 6. Right click on FFS05 and select Convert \rightarrow Change Template \rightarrow **Sandbox_Sugar_FFS** \rightarrow OK

☐ FF506	New	•	
🚉 Element Searches	Convert	•	🖷 Convert to Template
	🏷 🛛 Create or Update Data Reference		ৰাশ Convert to Model
	🕰 Categorize		🖫 Change Template
	Location		Change Reference Type
	Health		
Choose Element Templ	ate X		
Name:	FFS05		
Element Template:			
<none> Environment Production_FFS Production_Packagin Production_Sugar_F Sandbox_FFS Sandbox_Packaging Sandbox_Sugar_FF </none>	ng FS		
Templates of category: Warning: Changing the t have unintended conseq	<any> emplate of an existing Element may uences. Use with caution.</any>		
	OK Cancel		

Step 7. Select all the attributes of element FFS05 (you can select the first one, then keep Shift down and select the last one), right-click and choose "**Reset to Template**"

🖻 Categ	ory: Identification			
T	💷 Asset	N	Reset to Template	
	E Last Service Date	3	Create or Update PI Point	
	I Model	.	Categorize	-[]]
T	💷 Nominal Max Thre	- 1	Location of Element	
T	Product		Health of Element	
T	Production Line	\sim	Trend	

Step 8. Type in the name of the Model for FFS05 (we will come back to this in Section 2.4.5)

T	Last Service Date	1/1/2018 12:00:00 AM
	I Model	ADCO
	🗉 Nominal Max Throughput	75 Bags/min
	Product	Sugar

Excluded attribute property

Notice that the attribute list now matches what you see in the template configuration.

Also, when clicking on the General tab you'll see the AF element template used:

FFS05	_
General	Child Elements Attributes
Name:	FFS05
Descriptio	on:
Template	: Sandbox_Sugar_FFS

In order to exclude the attribute Moisture Content, not available for FFS05:

Step 9. Select the attribute **Moisture Content**, move to the right pane and select **Excluded** from the Properties dropdown menu.

More information on the Excluded attribute property is available in the LiveLibrary.

Name:	Moisture Content					
Description:						
Properties:	<none></none>	~				
Categories:	Configuration Item					
Default UOM:	Excluded					
Value Type:	Hidden					
Value:	Indexed					
Display Digits:	Manual Data Entry					
Data Reference:	Location 🕨	~				
	Health 🕨					

Step 10. Important: Click on

Check In to save your changes.

Step 11. Move back to **PI Vision**, open the display **Sandbox_FFS_Monitoring**, refresh the page and verify that FFS05 is visible under the Asset Dropdown Menu.

NB: It might take a few minutes for your PI Vision server to get updated with the latest changes on the AF Server (technically, PI Vision relies on the PI Web API Crawler to crawl the AF databases, and the crawling happens by default every 3 minutes. We lowered the crawling refresh interval to 1 minute for the purpose of this Lab).

Step 12. Verify what happens to the Moisture Content for element FFS05.



[it should show N/A, not available]

You might have noticed that each PI Point data reference attribute (i.e. each AF attribute pointing to a PI Point on the Data Archive, indicated by a \checkmark symbol) has a child attribute called *TagName*. Also, these attributes have a symbol next to them:

Category: Overall Equipment	Effectiveness
🖂 🗸 🖻 🔶 🎺 Availability	40.804 %
🗉 🧏 🗉 TagName	FFS01.Availability
🗆 🚽 🖉 🕈 🎺 Availability Mor	thly Average 72.766 %
TagName	FFS01.Availability Monthly Average
🗆 🗸 🗉 🔶 🎺 OEE	28.366 %
🗉 🦧 🗉 TagName	FFS01.OEE
🖂 🖉 🖻 🔶 🎺 Performance	96.552 %
🗉 🦧 🗉 TagName	FFS01.Performance
🖂 🖉 🖻 🔶 🎺 Quality	72 %
🗉 🧏 🗉 TagName	FFS01.Quality

Would you be able to tell (a) why these attributes show a symboland (b) why these attributes exist in the first place? This topic is a bit hairy, so we'll probably skip it during the Lab. Feel free to go through it at your own pace.

NERD ALERT - Hidden attribute property

The \boxed{R} symbol is used to indicate that these attributes are hidden.

A hidden attribute is one that cannot be retrieved by client applications such as PI Vision. This property is useful if an attribute is used to hold an **intermediate result** or a configuration parameter. For example, you could hide string values that are concatenated to build a tag name.

It can also be useful to set an attribute property to Hidden in a template when configuration has not been fully completed in the element itself. For example, elements are being created from a template with a PI Point data reference, but because some instrumentation is missing, PI Points do not yet exist for all the elements. By setting the attributes for the missing instrumentation to Hidden, a client application is prevented from obtaining an error result from a search (adapted from LiveLibrary).

In our case, the TagName attributes are hidden simply because they don't need to be available in PI Vision, as they are used for AF configuration purposes.

NERD ALERT - Referencing TagName child attributes

The TagName attributes have been used to store the name of the PI Point associated with the parent AF attributes (e.g. Availability, OEE, ...).

The reason is the following: suppose that many of your PI Points don't follow a clear naming convention. As an example, the Sealer Temperature of FFS01 could be associated with PI Point *FFS01SealTemp*, while the same attribute under FFS02 could be associated with PI Point *02ST3000*. Although this could seem illogical, it often occurs whenever the PI Points are named after the original data source items, which in turn follow the naming convention of the various PLC/SCADA manufacturers.

If you find yourself unable to establish a simple pattern that automates the associations between AF attributes and PI Points (e.g. using substitution parameters), you need to manually associate the AF attributes with the corresponding PI Points. The TagName child attributes allow you to "push" the PI Point names (e.g. *FFS01.OEE*) into the configuration of your PI Point data reference attributes (e.g. Sealer Temperature, Moisture Content etc.). To accelerate this operation, you could use **PI Builder**, an Excel Add-In that allows you to edit attribute configurations in bulk. For more information on using PI Builder, watch the playlist available on the <u>OSIsoft Learning channel</u>. In order to facilitate the system's maintenance, you can configure the TagName attribute as a Table Lookup, and store all PI Point names in an AF Table (more on AF Tables in Section 2.4.4).

NERD ALERT - Data Archive element

You might have noticed an AF element called *Data Archive* (we'll use italic font style to distinguish the element name from the actual Data Archive software component).

What is this element doing here, and why does it show an attribute containing the name of our Data Archive?

Elements	Data Archive	
Elements	General Child Elements Attributes Ports Analyses Notification Rules Version	
Data Archive		Grou
im im Production im im i	Filter 🔎 🔻	Name: Name
📖 武 Element Searches	Value ⊗	Description:
	Category: <none></none>	Properties: Hidden
	R Name PISRV01	Categories:
	L	Default UOM: <pre></pre>
		Value Type: String
		Value: PISRV01
		Display Digits: -5
		Data Reference: String Builder
		Settings
		%server%

The AF element *Data Archive* is simply a placeholder. It allows you to reference all the PI Point data reference attributes (e.g. Sealer Temperature, Moisture Content etc.) to the Data Archive where the PI Points reside. More importantly, if you need to **export the AF database and import it into another PI System**, you'll only need to modify the value of the Name attribute under the *Data Archive* element, and all the PI Point data reference attributes will point to the new Data Archive.

A typical use case is when you need to move a database from a production server to a sandbox server or vice versa. Most likely the Data Archive in the sandbox environment is named differently with respect to the Data Archive in the Production environment. Editing the value of the Name attribute under the Data Archive can save you a lot of time compared to editing hundreds or thousands of attribute configurations.

How is this behavior obtained? Take for example the Sandbox_FFS template and navigate to the Availability attribute. Click on Settings and inspect the "Data server" field (see red box in Figure 7). You'll notice that it points to the Name attribute under the *Data Archive* element using syntax:

%@\Data Archive|Name%

You'll notice also that the Tag Name field is pointing to the child attribute TagName, which we discussed before, using syntax:

%@.|TagName%

PI Point Data Reference									
Data server: %@\Data Archive Name% ~									
Tag name: %@. TagName%									
Tag Creation									
pointtype=Float64	•••								
O Attribute:	~								
Unit of Measure									
Source Units: <defa< td=""><td>ult> (%) <</td></defa<>	ult> (%) <								
Value retrieval methods									
By Time: A	Automatic 🗸 🗸								
Relative time:									
By Time Range: E	ind Time 🗸 🗸								
Calculation basis:	Time Weighted $$								
Min percent good:	80								
Preview									
Example instance: Select example instance									
Configuration:									
Value:									
Read only	OK Cancel								

Figure 7. PI Point data reference configuration using substitution parameters

For more information on different ways of pointing AF attributes to PI Points, please watch:

OSIsoft: PI Point data references Beginner to Advanced [v2.9.5.8368]

So far, we saw that one way of handling variations among assets is using the Excluded attribute property. This is useful in situations where not all attributes in an AF element template apply (in our example, the Moisture Content was not available for FFS05).

What happens when the **same variation** applies to **several elements** from the same element template? For example, if 5 out of 100 elements need an extra attribute, do I need to add the extra attribute to the element template used by all 100 elements and then exclude it from 95 of them? We will now see how derived templates help us handle this type of variations.

2.4.2. More on handling variations – Allow Extensions vs Derived Templates

Remember the second issue observed in PI Vision?



Sugar FFS Units									
Name 🔺	Trend	Average	Minimum	Maximum	StdDev	Units			
FFS03 Sugar Grain Size		0.49143	0	0.68254	0.29128				
FFS04 Sugar Grain Size		0.49143	0	0.68254	0.29128				
FFS05 Sugar Grain Size	J	0.54208	0	0.69801	0.26178	mm			
FFS06 Sugar Grain Size		0.49143	0	0.68254	0.29128				

Let's retrace how these AF attributes were configured in Sweet & Savory Corp's AF structure.

In order to comply with the company's quality standards, the **sugar** packed by the FFS units must meet strict requirements in terms of **grain size**. In out example, this is **not required for salt** packaging.

Allow Extensions

When creating the attribute Sugar Grain Size, the data management team enabled the option "Allow **Extensions**" (available under the General tab of the element template **Sandbox_FFS**):

Library	Sandbox_FFS						
Sweet & Savory Corp	General Attribut	General Attribute Templates Ports Analysis Templates Notification Rule Templ					
Implates	Name:	Sandbox_FFS					
Environment	Description:						
Production_FFS Production_Packaging	Base Template:	<none></none>			~	Type:	Ele
Here and Sandbox_FFS	Categories:				0	Default Attribute:	<
Sandbox_Packaging	Naming Pattern:					1	_
		Allow Ext	ensions	Base Template (Dnlv		
🗄 \cdots 📸 Transfer Templates	L						

This option allows you to add attributes to one or more individual AF elements without propagating the changes to all AF elements based on the same template.

Despite its **flexibility** and its usefulness **in testing phases**, an incorrect use of the Allow Extension option often leads to a **lack of consistency** among elements, and results in manual operationswhenever the extra attributes need to be added to another element.

In our specific case, a mistake was made in the attribute configuration: the Sugar Grain Size attributes of units FFS03, FFS04 and FFS06 are all associated with the same PI Point on the Data Archive. In other words, all attributes Sugar Grain Size are looking at the same source data, the data for FFS03. This is a very common "copy-paste" typo!

Derived Templates

A better way to create additional attributes for one or more elements is to use **Derived Templates**.

Expand Sandbox_FFS (our base template), you will see a derived template called Sandbox_Sugar_FFS.



By default, the attribute templates tab of derived templates shows only the attributes that do not exist on the base template, or attributes for which we want to override the configuration compared to the base template.

Once you select a derived template, you can display also the attributes related to its Base Template by ticking **Group by: Template**:


In our case, the derived template includes an additional attribute called **Sugar Grain Size**. We can use this derived template instead of recurring to the *Allow Extensions* option.

Step 1. Go to the Elements tab, navigate to Sandbox \rightarrow Packaging \rightarrow **FFS03**

Step 2. Right click on FFS03 and select Convert → Change Template → Sandbox_Sugar_FFS

🖮 🗊 Sandbox					
🗄 🗝 🗊 Packaging					
🗇 FFS01					
🗇 FFS02					
🗇 FFS03				1	
🗇 FFS04		New	•		
🗇 FFS05		- ·			
🗇 FFS06		Convert	- F	* *	Convert to Model
🔍 Element Searches	а,	Create or Undate Data Reference		E.	Change Tangalata
	~	create of opdate bata Reference		40	Change Template
	S.	Categorize			Change Reference Type
		Location			5 51

Choose Element Temp	late	×						
Name: Element Template:	FFS03							
<none> Contemport Contempo</none>	ing FFS 9 %S							
Templates of category:	<any></any>	\sim						
Warning: Changing the template of an existing Element may have unintended consequences. Use with caution.								
	OK Cancel							

We now need to reset the Sugar Grain Size attributes to the template configuration. This is necessary to force AF to remap the attribute to the correct PI Point!

Step 3. Click on Search \rightarrow Attribute Search...



Step 4. Use as filters:

- Attribute Name: Sugar*
- Search Root: Sandbox

Attribute Search								-		×
Server:	9						V III Conn	ort		
								~		
Database:		Sweet & Sa	vory Corp				~ ···			
Where Attribute name:		*Sugar*						~		
Attribute descrip	tion:							_		
Attribute catego	ry:	<anv></anv>						~		
Attribute value t	ype:	<anything></anything>						\sim		
Maximum results	:							1000		
Element Criteria										
Search Root:	Sandhox							Searc	h Sub-Elem	ents
Name:								1		
Description:]		
Catagoriu	ZAIIN								Search	\sim
Category:	SAIL2						•			
Template:	<aii></aii>						~			
Type:	Any						~			
Search results:	Т	ne search foun	d 4 Attributes matching t	he sear	ch criteria.					
	s	weet Savory	Corp					Group	by: 🗹 Ca	tegory
Data Archive	F	ilter								<mark>ب م</mark>
	Г	🖊 : 🗉 🞗	Name	۵	Value	Path	I			(à
		🛛 🖻 Categ	jory: <none></none>							
			🎺 Sugar Grain Size	¢	0.65622 mm	Sand	box\Packaging\FFS0:	3 Sugar Grai	n Size	
			🍼 Sugar Grain Size	¢	0.65622 mm	Sand	lbox\Packaging\FFS06	5 Sugar Grai	n Size	
			🎺 Sugar Grain Size	C	0.65108 mm	Sand	lbox\Packaging\FFS0	5 Sugar Grai	n Size	7///
			🍼 Sugar Grain Size	9	0.65622 mm	Sand	lbox\Packaging\FFS0+	1 Sugar Grai	n Size	777
							OK	Cancel	Da	eset

Step 5. Select the Element Attribute Search Results (as in screenshot below) and select all the

attributes; right-click and select Reset to Template.



Elen	nent A	ttrik	oute Search Results 3			
Filte	r					
	/ :	• 8	Name	V	/alue Pa	at
	🖻 (ate	gory: Process Data			
Ð	0	T	🍼 Sugar Grain Size	5	Reset to Template	4
Ð	0	T	🎺 Sugar Grain Size		- W	-
Ð	0		🎺 Sugar Grain Size	7	Create or Update PI Point	_
Ð	0	T	🎺 Sugar Grain Size		Categorize	

You should see that the Sugar Grain Size now shows the correct PI Point (for example, FFS04 will have its Sugar Grain Size attribute pointing to the PI Point *FFS04.Sugar Grain Size*).

FFS0	4											
Gen	eral	Child El	ements	Attributes	Ports	Analyses	Notificat	tion Rule	s V	ersio	n	
	Group by: 🗹 Category											
Filte	97							Q	•		Name:	Sugar Grain Size
		= \$ \$	Name				۵	Val	^		Description:	
	Ē						Properties:	<none></none>				
	0		Of	berator				Mary			Categories:	Process Data
⊞	0		🍼 Pa	cking Rate P	er Minut	e		58 B		Def	Default UOM:	millimeter
⊞	0	∎ 🔶	🍼 Pa	cking Rate S	itatus			Far		,	Value Type:	Double
		T	🔳 Pa	cking Target	t per Min	ute		60 B		1	Value:	0.67349 mm
∥⊞	0	T	Ø Pr	oduct Bag Si	ze			220 g			Display Digits:	-5
∥⊞	4	T	🎺 Running State					1			Data Reference:	PI Point
∥⊞	4	T	🧭 Sealer Temperature 57.7									Settings
⊞	4	T	Ø SH	iift				Shift 2			VPISRV01VFFS0	4.Sugar Grain
Œ	0	T	🍼 Su	ıgar Grain Siz	ze			0.67			Size";UOM=mm;p	ointtype=Float64

Step 6. Move between elements and make sure that all attributes are populated with live data. Again,

you'll notice a small symbol next to each attribute, which means that the attribute is templatized.

Step 7. Move to the Library tab, select template Sandbox_FFS (Figure 8).

Step 8. Under the General tab, untick the "Allow Extensions" checkbox (this will prevent accidental/undesired creation of attributes at the level of a single element)

Library	Sandbox_FFS							
Sweet & Savory Corp Templates Completes 	General Attribu	te Templates Ports Analysis Templates Notification Rule Templates						
	Name:	Sandbox_FFS						
	Base Template:	<none> V Type:</none>						
	Categories:	Default Attri						
	Naming Pattern:							
⊕… 蹭 Model Templates ⊕… 谙 Transfer Templates		Allow Extensions Base Template Only						
Enumeration Sets		Extended Properties (1) Location Health Security						
Em Tables	Find:	Derived Templates Elements Referenced Parent Templates						
🛄 Model Specifications		Derived Elements Referenced Child Templates						

Figure 8. Disable Allow Extensions

Step 9. Important: Click on	💐 Check In	to save your changes.
Step 9. Important: Click on		to save your changes.

Step 10. Move to PI Vision → display Sandbox_Overview_FFS

Step 11. Refresh the page and verify that now the table shows different trends and values for each Sugar Grain Size attribute.

Sugar FFS Units

Name 🔺	Trend	Average	Minimum	Maximum	StdDev	Units
FFS03 Sugar Grain Size		0.45768	0	0.69904	0.3033	mm
FFS04 Sugar Grain Size	1 m	0.45843	0	0.69426	0.29282	mm
FFS05 Sugar Grain Size		0.52984	0	0.68985	0.2696	mm
FFS06 Sugar Grain Size		0.60968	0	0.68197	0.19692	mm

	Always use AF templates.					
V	If one or more assets differ from the others, you can leverage Derived Templates					
Best Practice	(and/or the Excluded attribute property) to handle these variations.					
	We recommend caution when using Allow Extensions , as the additional attributes					
	created with this option must be manually configured for each "extended" element.					

2.4.3. Units of Measure

The UOM database provides automatic handling of simple conversions between units of measure for attributes of the same UOM class. A UOM class is defined by the fundamental dimensions of its measurement. Examples of UOM classes are Mass, Volume, and Density. The UOM database comes preloaded with numerous standard unit-of-measure classes and conversion factors. You can extend these classes by adding new units of measure, as well as new measurement classes. The implementation of UOM is based on the International System of Units (SI).

While navigating in PI Vision, we noticed that the Sealer Temperature was out of the expected range (it should be between 70 and 270 °F).



Issue # 3. The attribute Sealer temperature shows values outside the expected range.

Since this issue occurs for all FFS units, it's not likely due to a sensor's malfunctioning! Let's verify that the AF attributes have been correctly configured.

Sweet & Savory Corp has bought a stock of temperature sensors from Europe, so **the temperature is actually measured in degrees Celsius**. However, we expect the AF Server to provide us the values in ^oF while handling the UOM conversion behind the scenes. Let's make sure that AF is aware of the UOM of the source data (Figure 9).

Step 1. In PSE, move to the Library tab \rightarrow **Sandbox_FFS** \rightarrow Attribute Templates tab

- Step 2. Select the Sealer Temperature.
- Step 3. Click on Settings...
- **Step 4.** Select explicitly **C** instead of the default value (in this case *<Default>* (*PF*)).

Library	Sand	lbox_F	FS										
🗳 Sweet & Savory Corp 🔹 🔺	Gen	eral 4	Attribute Template	es Ports	Analysis Te	emplates Not	ificatio	n R	ule Templates				
🖃 🗝 Templates											Group by:	Category 🗹 T	emplate
Element l'emplates	Filte	r					ν		Name:	Sealer Ten	nperature		
G Production_FFS		/ i	A R Name		4	Description	3 ^	-	Description:				
····· 🔂 Production_Packaging			X Hanc			Description	-		Properties:	<none></none>			~
Bandbox_FFS			Kan Moistur	e Content									
Sandbox_Packaging	11		- Nomina	Max Thro	uabout		-		Categories:	Process Da	ata		
🗄 ···· 📽 Model Templates	11_				ugriput		-		Default UOM:	degree Fa	hrenheit		~
🗄 📸 Transfer Templates	⊞		OEE				_		Value Type:	Double			\sim
⊕ └on Sets ⊕ ∀ Reference Types	⊕		K Operat	or					Default Value:	0 ºF			
Tables	⊞		Kara Packing	Rate Per I	Minute				Display Digits:	-5			
Model Specifications	⊞		🕈 🛛 🍊 Packing	Rate Stati	us				Data Reference:	PI Point			\sim
III Product Assignment	1		📑 Packing	Target pe	r Minute					3	Settings		
Sandbox_FSS_Maintenance	⊞		🔶 🍊 Perform	ance				Þ	\\%@\Data Archiv	ve Name%\	%@. TagName%;p	ointtype=Float64	
Table Connections	1		📑 Produc	t								_	
Analysis Categories	⊞		K Produc	Bag Size		PI Point D	ata Re	efei	rence		×		
Attribute Categories Element Categories	1		📑 Produc	ion Line		Data serv	/er:		%@\Data ArchiveII	Name%	~		
Notification Rule Categories	⊞		🔶 🛛 🍊 Quality			Tag n	ame		%@.lTagName%				
Elements	⊞		Kunnin 🦝	g State			Tag	Cre	ation				
Event frames	Œ		🍊 Sealer	Femperatur	re 2	p	ointtyp	e=	Float64				
	Œ		K Shift										
unit of Measure	11		Throug	hout Efficie		🔿 Attrib	ute:				~		
A Contacts	11		i lan ang ang ang ang ang ang ang ang ang a	ipor enice	льсу	Unit of I	Measu	re					
💥 Management	<					Source	Units:		<default> (°F,</default>	~			
Sealer Temperature									<default> (°F)</default>		4		i
Sealer Temperature		_				Source Value re	Units:	Ime	<default> (°F) <default> (°F)</default></default>	(4		

Figure 9. Setting the Source UOM.

Step 5. Click OK and verify that the **Default UOM** is still ^oF (Figure 10). The **Default UOM** configuration field defines the unit of measure used **to display the attribute's values on client applications**.



Figure 10. Default UOM vs Source Unit

Step 6. Go back to PI Vision, refresh the display and verifythat now the values fall within the expected operational range.



You may be wondering... and what about the Data Archive? Aren't UOMs defined for each PI Point as well?

NERD ALERT – Data Archive engineering units vs Asset Framework UOMs

click the "PI Point Properties" button [[[]] (Figure 11).

You may run into a situation where you need to set the UOM of an AF attribute and you want to inspect the settings for the PI Point. The easiest way to do this is to select the attribute within an element and

🛨 🧳 🗉 🎺 Se	ealer Temperature -17.3	778 ⁰두							
PI Point Data Reference	:e	×	🚰 Pl Point Prop	erties		—		\times	
			PI Point						
Data server: P	ISRV01		Name:	VPISRV01	FFS01.Sealer Temperature				
Tag name: Ff	FS01.Sealer Temperature		Current Value:	-17.77777	-17.777777777778 2/6/2020 2:51:00 PM				
O Attribute:		~	Time Stamp:	2/6/2020 2					
Unit of Measure			Alphabetic	L					
Source Units:	<default> (°F_ ~</default>		Name		Value			^	
Value retrieval metho	ds		datagroup		piadmins				
By Time:	Automatic	~	dataowner		piadmin	piadmin			
			datasecurit	ty	piadmin: A(r,w) piadmins: A(r,w) PIWorld: A(r)				
Relative time:			descriptor		<empty></empty>				
			digitalset		<empty></empty>				
By Time Range:	End Time	~	displaydigit	s	-5				
Calculation basis:	Time Weighted	\sim	engunits		deg Celsius				
			excdev		0.1				
Min percent good	l• 80								

Figure 11. The PI Point Properties window

Remember: **the** *engunit* **field is optional** when configuring a PI Point (just as the UOMs are optional for AF attributes), so you might not find the information you are looking for. Also, the *engunit* is **simply a character string**, it is not involved in UOM conversion and it does not respect the UOM terminology used in the AF Server (degrees Celsius can be expressed as ^oC, or as deg C, or in any other way). It is discretion of the Data Archive administrator to correctly configure the *engunit* field of the PI Points.

A common situation you may run into is that the attribute is resolving the PI Point using a replacement character string such as %@. |TagName%. If this is the case, when clicking on the PI Point properties

button

you'll probably get an error popup such as:

Cannot retrieve PI Point '<PI Point name>' for attribute '<path to AF attribute>'.



Before we can view the properties of that PI Point, we need to force AF to resolve the PI Point name and store it in the attribute configuration. This can be done by right-clicking on the attribute and selecting *"Create or Update PI Point"*.



Figure 12 shows the same AF attribute, before (left) and after (right) performing the "Create or Update PI Point" operation.

Catego	ory: Process Data			C] Catego	ry: Process Data	
	🍼 Bags Today	33894 Bags			T	🍼 Bags Today	33894 Bags
0 🗉	🎺 Moisture Content	0.088093 %		1	T	🎺 Moisture Content	0.098591 %
🔳	R TagName	FFS11.Moisture Content				R II TagName	FFS11.Moisture Content
0	PI Point Data Reference		×	0	T	PI Point Data Reference	e X
ø 🗉				0			
ø 🗉 🔶	Data server:	~		0	•	Data server: PI	ISRV01 v ···
	Tag name: %@	. TagName%	\triangleright		T	Tag name:	FS11.Moisture Content
ø 🗉	O Attribute:		~	0	T	Attribute:	~
0 🗖	Unit of Measure			3	T	Unit of Measure	
ø 🗉	Source Units: <	Default> (% 🗸	ă	0		Source Units:	<default> (% ∨</default>
0	Value retrieval methods			3		Value retrieval metho	ds
	By Time:	Automatic	~			By Time:	Automatic \checkmark
	Relative time:		51			Relative time:	
	By Time Pange:	End Time	~			By Time Range:	End Time 🗸
	Calculation basia	Time Weighted				Calculation basis:	Time Weighted
	Calculation basis:	Time weighted	<u> </u>			Calculation basis.	
	Min percent good:	80				Min percent good	: 80
	Read only					Read only	
		OK Cancel					OK Cancel

Figure 12. Tag name resolution before and after the "Create or Update PI Point" operation.

When this action is carried out, AF resolves the replacement characters and locks in the settings. This is often done to improve performance in larger systems as the client tools will no longer have to use processing power or network bandwidth performing the conversion from a replacement character string to the actual tag name. This efficiency improvement is negligible on a small system but becomes relevant for large, corporate-level systems. The drawback is that, in case the value of attribute *TagName* changes, the PI Point data reference attribute does not get automatically updated. The solution in this case is resetting the attribute configuration to the template, and then performing again the "Create or Update PI Point" operation.

For more information on what the "Create or Update PI Point" operation does, please read <u>Attribute</u> <u>indicators for updates of PI point data references</u>.

2.4.4. AF Tables

Tables are held in the AF database to provide contextual information through the Table Lookup data reference. Tables can provide information about the equipment or process entities; also, they can be used to store tag names or other configuration information.

Tables can be created internally, imported from an external relational or tabular data source, or linked dynamically to an external relational data source. In this way, Tables can expose information in maintenance, production planning, or equipment databases for use by PI client applications.

When you opened PSE for the first time and navigated to the sandbox elements, you probably noticed that it took a long time to populate the attributes' values in the Attributes tab. We will see that this slowness is caused by a single heavy hitter, the attribute **Last Service Date**. This attribute is a Table Lookup, which means that it... looks up a value in a table, using a SQL-like query of type:

SELECT *column* FROM *table* WHERE *WhereClause* ORDER BY *column* ASC|DESC;

Let's look at the table we use to populate the Last Service Date attribute:

Step 1. In PSE, move to the Elements tab \rightarrow Sandbox \rightarrow Packaging \rightarrow FFS01.

Step 2. Open the Attributes tab and select the Last Service Date attribute.

Step 3. Click on Settings...

Step 4. Take note of the table used(this table appears also in the SQL query under the Settings button)

SELECT [Last Service Date] FROM [**Sandbox_FFS_Maintenance**] WHERE Asset = @Asset ORDER BY Last Service Date

Elements		FFS01							
🖃 뤔 Elements		Gener	ral Child E	lements Attributes Ports A	nalyses Notification	Rules	Version	Group by: 🔽 Category	Template
🕂 🗇 Production		Filter			-	Name:	Last Service Date		
B Packaging		✓ : ■ ♦ & Name △ Value ③			^	Description:			
🗇 FFS01		Category: Identification			1	Properties:	<none></none>	~	
🗇 FFS03		🔳 🗉 Asset		FFS01		Categories:	Identification		
🗇 FFS04			T	I Last Service Date	11/1/2019 12:00:	Default UOM: <none></none>		<none></none>	
FFS06			E	I Model	ADCO		Value Type:	DateTime	
Element Attribute Search F	Results		T	Nominal Max Throughput	75 Bags/min		Value:	11/1/2019 12:00:00 AM	
					Salt		Display Digits:	-5	
Table Lookup Data Ref	erence	11		- House	X		Data Reference:	Table Lookup	~
Table:	Canally		Matalana				1	Settings 3	
Beer the set of set	Sandb	0X_F55		4		•	SELECT [Last Ser	vice Date] FROM [Sandbox_FSS_Ma	intenance]
Result column:	Last Se	ervice D	ate				WHERE Asset = (@Asset ORDER BY Last Service Date	
Unit of Measure:	<none< td=""><td>:></td><td></td><td></td><td>~</td><td></td><td></td><td></td><td></td></none<>	:>			~				

Step 5. Move to the Library tab and open the Tables dropdown list.

Step 6. Select table Sandbox_FFS_Maintenance.

Step 7. Move to the Table tab.



How many rows are present in this table? [hint: far too many!]

Do we need all these rows? [hint: no, we only need the FFS Units that appear in our database!]

The table Sandbox_FFS_Maintenance is stored in a central SQL Server at the corporate level and contains asset metadata from all the production sites around the world (around 300 000 FFS units... probably unrealistic, but good for our discussion!). For our goals, we only need to retrieve the data for our pilot site. This has already been done correctly in the Production environment, so we just need to understand how it works and replicate it in the sandbox environment.

Step 8. Still on table Sandbox_FFS_Maintenance, move to the **General** tab.

Step 9. Click on the "**Link...**" button. You'll see a Connection String and a query to an external SQL Server. The current query is retrieving all rows and columns from the external SQL table:

select * from dbo.PackagingPlantMaint

Table Link		Х
Name:	Sandbox_FFS_Maintenance	
Description:		
Connection:	ISRV01;Use Encryption for Data=Fals	e;Tag with column collation when possible=False; \checkmark
Query:	select * from dbo.PackagingPlantMair	nt
Parametere	Parameter Name	Default Value
raianeters;	@	Add

Step 10. Click on **Cancel** and select table **Production_FFS_Maintenance**.

Step 11. Move to the Table tab, how many rows do you see?

NB: The Table tab shows only one row corresponding to the default value of the @Asset parameter (FFS01), but the query will work fine for any other AF element.

Library	Product	ion_FFS_Main	tenance	
Sweet & Savory Corp	General	Table Defi	ine Table Version	
🖃 🐨 🐨 Element Templates	Product	ion_FFS_Maint	enance	
Event Frame Templates	Filter			
Model Templates		Asset	Model	Last Service
		Asset	Hodel	Date
🖅 🗠 🗠 Reference Types		FFS01	ADCO	11/1/2019 12:0
E Tables				
III Model Specifications III Packaging Target				
Product Assignment Production_EES_Maintenance				
Sandbox_FFS_Maintenance				

Step 12. Go to General \rightarrow "Link..." and check how the query is performed. You'll notice that the SQL query contains a parameter (*where @Asset = Asset*). You can hit Cancel to close the window.

Table Link			×
Name:	Production_FSS_Maintenance		
Description:			
Connection:	ISRV01;Use Encryption for Data=Fals	e;Tag with column collation when pos	ssible=False; 🗸
Query:	select Asset, Model, [Last Service Da where @Asset = Asset	te] from dbo.PackagingPlantMaint	
	Parameter Name	Default Value	
Parameters:	@Asset	FFS01	<u>Delete</u>
	@		Add
Security:	Impersonate Client Supply Password Change Pas No additional security context	sword	
		OK	Cancel

Using parameters in a linked table query is useful to limit the number of rows returned from a very large external table. You can add conditions and parameters to return more targeted results, such as all rows that include a device or manufacturer ID number, specific for each table lookup data reference. (LiveLibrary).

We will now go back to the Sandbox table and edit the query so that only needed information is returned (Figure 13):

Step 13. Open table **Sandbox_FFS_Maintenance** → General tab.

Step 14. Select the Link... button.

Library	Sandbox_FSS_Maintena	ance
Sweet & Savory Corp	General Table Defin Name: S	e Table Version andbox_FSS_Maintenance
⊕····	Description: Categories:	inked - Provider=SOLOLEDB.1;Integrated Security=SSPI;Persist Security Info=False;Initial Cata
	Query: Li Time Zone: <	inked - select * from dbo.PackagingPlantMaint
Tockgary raiged T	Cache Interval:	1 Hours lecurity Import Link Unlink
Calegories	Table Link Name: Description: Connection:	Sandbox_FSS_Maintenance SBU(1):Lise Encountion for Data_Ealse:Tag with column collation when possible=Ealse: X
	Query:	select * from dbo.PackagingPlantMaint

Figure 13. Link configuration tab of table Sandbox_FFS_Maintenance

Step 15. Edit the query as follows (feel free to copy and paste from the .PDF file of this workbook):

Select * from dbo.PackagingPlantMaint

where @Asset = Asset

Step 16. In the Parameters Section, add the parameter @Asset (see also Figure 14):

Parameter Name	Default Value
@Asset	FFS01

Table Link					×
Name:	Sandbox_FSS_Maintenanc	e			
Description:					
Connection:	Provider = SQLOLEDB. 1; Int	egrated Securi	ity=SSPI	;Persist Security I	nfo=False;Initial Ca 🗸
Query:	select * from dbo.Packagii Where @Asset = Asset	ngPlantMaint			
	Darameter Name		Default	Value	
Parameters:		EE	S01	value	Delete
	@	11	301		Add
Security:	 Impersonate Client 				
	O Supply Password	hange Passwo	ord		
	○ No additional security c	ontext			
				OK	Cancel

Figure 14. Modifications to the Sandbox_FFS_Maintenance table

Step 17. Click on and open the Table tab: how many rows are present now?

General	Table	Define Table	Version				
Packer	PackerMaintenance						
Filter							
	Asset	М	lodel	Serial Number	Installation Date	Last Service Date	Service Crew
•	FFS01	AD)CO	ADA5FFS01	5/1/2017 12:00:00 AM	11/1/2019 12:00:00 AM	Red

Only one row is returned, and it corresponds to the Default value of the SQL query.

NB: as stated before, the parameter value in the configuration above is just an example parameter value (a default value) and it doesn't affect the query when evaluated for a different AF element.

Step 18. Move to the **Sandbox_FFS** and select the Attribute Templates tab.

Step 19. Select the Last Service Date and click on Settings...

Step 20. Add the Table Parameter as in the screenshot below (Parameter @Asset = Value @Asset).

Library	Sandbox_FFS			
Library	Sandbox_FFS General Attribute Templates Ports Filter RName Template: Sandbox_FFS Availability	Table Lookup Data Refe Table: Result column: Unit of Measure: Time Zone: Behavior	eference Sandbox_FSS_Maintenance Last Service Date	
Transfer Templates Finance Transfer Templates Finance Types Tables Model Specifications Packaging Target Product Assignment Product on_FSS_Maintenance	Image: Availability Monthly Average Image: Availability Monthly Average Image: Bags Today Ima	Rule: Order by: Where Column: Asset ~	Select first row matching criteria Last Service Date Operator: Attribute or Value = V	
Categories Analysis Categories G Analysis Categories G Analysis Categories G Analysis Categories G Attribute Categories G Element Categories	Image: Nominal Max Throughput Image: OEE Image: OEE Image: Operator	Complete WHERE Clause: Asset = @Asset		
Intification Rule Categories	Image: Constraint of the second se	Table Parameters Parameter @Asset	Value @Asset	
🕌 Library	A Performance			

In this way, every time a value is requested by a Table Lookup attribute, the parametrized query wil be passed to the external SQL server and only the requested value will be returned.

Step 21. Important: Click on



Check In to save your changes.

Step 22. In order to verify whether the caching performance has improved, close PI System Explorer and PI Vision. Then reopen PI System Explorer and move between AF elements under the Sandbox branch (make sure you have the Attributes tab open). Do you notice any improvements in terms of data retrieval speed? [hint: yes!:D]

If you like, you can further limit the number of results returned by the SQL Query. What we needfrom the external SQL table are just the Asset name and the Last Service Date. We don't need to import also the Model, Serial Number, Installation Date and Service Crew. We can modify the query of the Sandbox_FFS_Maintenance table as follows:

Service from dbo.PackagingPlantMaint select Asset, [Last Date] where @Asset = Asset

NB: Don't forget the **square brackets** around Last Service Date. The space characters in the column name Last Service Date would break the SQL query unless square brackets are used.

Still hungry for knowledge on AF Tables? For more information on parametrized queries, please visit <u>Data</u> references from outside the PI System and Parameters for linked table queries.

Also, this article covers how AF Tables handle local caching: How AFTable Caching Works (4).

Last but not least, this PI Square post illustrates good and bad uses of Table Lookup, Formula and Analysis data references: <u>Asset Analytics Best Practices - Part 3: Input Attributes (</u>5).



Before we move on, let's look at what attributes could slow down data retrieval, and what actions you can take to solve these performance issues.

NERD ALERT – Heavy hitters

Table Lookup

Table Lookup attributes that point to externally linked tables are the usual suspects when you experience slow data retrieval. The performance issue can be caused by the size of the table (as in our example), by network latencies between the Asset Framework server and the external data source, by an overloaded external data source etc.

As you see, not all these issues can be directly addressed in Asset Framework. Parametrized queries can drastically reduce latency when the main issue is the table size; another option is importing the table in AF (in this case, it is a good practice to limit your imported tables to 10,000 rows of data or less).

Formula

Formula data reference allows you to perform simple calculations that are evaluated on-the-fly by the client application (e.g. PI Vision, PI DataLink). Performance issues can arise in case of long dependency chains (i.e. when a Formula needs to wait for its input values coming from a chain of other Formulas).

The best solution is to move the calculation to Asset Analytics and use PI Points to store the calculation results.

Summary calculation via Value Retrieval Methods

Value Retrieval Methods, available for the PI Point data reference, allow you to perform statistical calculations (such as an average or a total) over a desired time range. This calculation is requested on-the-fly by the client application and performed by the PI Archive Subsystem on the Data Archive.

In the example in Figure 15, we request a 90-day average for PI Point BA:ACTIVE.1. The evaluation time will depend on the data density (the number of events for that PI Point and for the requested time range) and the available resources of the Data Archive.

Again, the best solution is to move the calculation to Asset Analytics and use a PI Point to store the calculation results.

For more information on Value Retrieval Methods, please refer to <u>Configuration of retrieval methods for</u> <u>attribute values</u>.

PI Point Data Referen	ice	×	
Data server:	PISRV01	~ ••	
Tag name:	BA:ACTIVE.1	m	
O Attribute:		~	
Unit of Measure			
Source Units:	<none> ~</none>		
Value retrieval meth	ods		
By Time:	Automatic	\sim	
Relative time:	-90d		Average ~
By Time Range:	Average	\sim	Count
Calculation basis	Time Weighted	~	End Time
Min percent goo	d: 80		Minimum Deside Keel And Device Keel
			Range
			Standard Deviation
	ОК	Cancel	Start Time Total

Figure 15. Value Retrieval Methods for the PI Point data reference (left) and available functions (right)

2.4.5. Enumeration Sets

An enumeration set is an ordinal list of sequential integer values, which are mapped to names. It allows AF attributes to refer to a common term instead of a number. You typically use enumeration sets to establish predefined values for attribute templates. When you configure AF attributes based on those templates, you have users select those values frompre-populated lists rather than typing valuesmanualy. This helps ensure you have consistent nomenclature throughout your database.

Remember what happened when you opened **Sandbox_FFS_Monitoring** and switched to FFS04?

Issue # 5. Asset FFS04 shows "No Data" for the Nominal Max Throughput

Name	Value	Units
FFS04 Model	MF TECHNO	
FFS04 Shift	Shift 2	
FFS04 Operator	Toni	
FFS04 Nominal Max Throughput	No Data	Bags/min
FFS04 Packing Rate Per Minute	53	Bags/min
FFS04 Last Service Date	11/1/2019 12:00:00 AM	

Let's see where this issue leads us to!

Step 1. In PI System Explorer, under the Sandbox branch select FFS04.

Step 2. Select attribute **Nominal Max Throughput** and click on **Settings...** As we saw in exercise 2.4.3, this attribute is of type Table Lookup, and retrieves its value using a SQL-like query.

Attribute **Model** is used in the configuration of attribute **Max Throughput** as input parameter of the WHERE clause:

Table Lookup Data Refe	rence		×
Table:	Model Specifications		v 🚥 🚰 🛅
Result column:	Max Throughput		V Stepped
Unit of Measure:	evt/min		\sim
Behavior			
Rule:	Select first row matching criteria		~
Order by:	<none></none>		← ASC ···································
Where			
Column:	Operator: Attribu	ute or Value:	Add And
Max Throughput 🗸 🗸	= ~ @Ass		Add Or
Complete WHERE Clause:			
Model = @Model			^
			~

Let's look at the AF Table that this attribute retrieves its value from.

Step 3. Move to Library \rightarrow **Tables** \rightarrow **Model Specifications** \rightarrow Table tab



We notice that the two models are ADCO and MF TECNO, while the value entered in the Model attribute for FFS04 is MF TECHNO. The extra "H" causes the Nominal Max Throughput attribute to fail to resolve, and the Throughput Efficiency fails as well, since it uses the Nominal Max Throughput as a formula input.

In order to solve this issue and to **avoid future typos**, we need to edit the template **Sandbox_FFS** so that the attribute **Model** uses an Enumeration Set. We want to use the Enumeration Set **Model**.

Library	Model List			
Sweet & Savory Corp	General			
Templates Element Templates Event Frame Templates Model Templates Transfer Templates Transfer Templates Model List Reference Types	Name: Model List			
	Description:			
	Hexadecimal Security			
	Value 🔺 Name			
	0 MF TECNO			
Tables	▶ 1 ADCO			

Step 4. Go to the **Library** tab and select **Element Templates** \rightarrow **Sandbox_FFS**.

Step 5. Move to the Attribute Templates tab and select the attribute Model.

Step 6. Edit the Value Type and choose **Enumeration Sets** \rightarrow **Model**.

Library	Sandbox_FFS		
Sweet & Savory Corp	General Attribute Templates Ports	Analysis Templates Notificatio	n Rule Templates
🚊 🔽 Templates			Group by: 🗌 C
Element Templates	Filter		Model
	R Name 4	≥ ©sa ^ Description:	
Production_Packaging Immediate Grant Standbox_FFS 4	🗉 🔂 Template: Sandbox_FFS	Properties:	<none></none>
🔤 🖓 Sandbox_Packaging	Asset	Categories:	Identification
	Availability	Default UOM:	<none></none>
🗄 🕀 Transfer Templates	Availability Monthly Average	Value Type:	String
Enumeration Sets		Default Value:	Basic Types
Em the Reference Types	Bags Today	Display Digits:	Array Types
Tables	🔄 Last Service Date		Enumeration Sets Model List
III Model Specifications	Model 5	Data Reference:	Objects b System
Eackaging Target			Setungs
Product Assignment	Moisture Content		

Step 7. Move to the Elements tab and specify the model of your FFS units under the Sandbox environment. Refer to Table 3 (next page) to choose the correct model for each element.

Check In to save your changes. Step 8. Important: Click on

Step 9. Move to PI Vision → display Sandbox_FFS_Monitoring, refresh the page and verify that now the table shows good values for the FFS04 attribute Nominal Max Throughput.

Elements	FFS01
Elements PI Data Archive Production Constraints Production Constraints Production Constraints Production Productin Production Production Production Production Produc	General Child Elements Attributes Ports Analyses Notification Filter Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Identification Image: Category: Image: Category: Identification Image: Catego

Table 3. Models associated with FFS units 1 to 6:

FFS unit	Model
FFS01	ADCO
FFS02	ADCO
FFS03	ADCO
FFS04	MF TECNO
FFS05	ADCO
FFS06	MF TECNO

Note: Starting with **PI Server 2018 SP2**, **Enumeration Sets can be created from Digital Sets** (stored in the Data Archive) using PI System Explorer. This helps you ensure consistency between the Data Archive and the Asset Framework server. For more information, refer to <u>Create enumeration sets from digital state</u> sets and review digital state sets on a PI Data Archive.



Use Enumeration Sets to establish predefined values for attribute templates. This ensures you have consistent nomenclature throughout your database.

2.4.6. NERD ALERT - String Builder

The String Builder data reference enables you to apply string manipulation functions, such as concatenation, to attributes' values, and output a reformatted string. This is useful when you need to obtain a string or numeric value type from other element attributes.

Below are some applications of the String Builder data reference:

- ✓ Concatenate strings (Element Name + Attribute Name + Value).
- \checkmark Format dates and numbers.
- \checkmark Build paths to elements and attributes.
- \checkmark Parse comments from operators stored in PI Points.
- \checkmark Display element information as an attribute.

For more information please visit: String Builder data references.

Let's now see it in action. Remember the last issue we observed in section 2.3?

Issue # 6. The title of the display Sandbox_FFS_Monitoring_Faulty_Title does not update with the chosen asset



The title "FFS04 Sugar Packaging" was built using static text boxes, hence it doesn't update when changing asset. By hovering over any of the symbols we see that the process data has updated with the correct asset (e.g. FFS01), but the title still shows FFS04.

In order to make this title automatically update with the chosen asset, we need to use the AF attributes **Asset** and **Production Line**. These attributes are configured using the String Builder data reference, and point to the name of the asset itself and its parent asset. Let's first create a dynamic title and then we'll look behind the scenes at how String Builder attributes can retrieve element's names.

Step 1. Switch to Edit mode (button in the top-right corner of the display) and delete the existing title. You can simply select the title and hit the delete key.

Step 2. Open the asset pane on the left side of the display

Step 3. Navigate to **Sandbox** \rightarrow **Packaging** \rightarrow **FFS01**.

Step 4. From the symbol pane, select the Value symbol in the top-left corner **select**, then drag and

as in Figure 16).

drop the attribute Asset on the display (you should see this symbol

Assets Ad Hoc Display N 🗆 🔻 T 🔛 R ۹ FFS04 Sugar Packaging back to overview FFS0 O FFS03 123 G FFS04 PFS05 O FFS06 Attributes dentificati Asset Last Service Date Mod Nominal Max Throughpu Product

Figure 16. Creating a dynamic title in PI Vision

The symbol will include the name of the attribute, its current value and the timestamp. In our case, we only need the value field.

Step 5. Right click on the symbol and select Format Value.

Step 6. Expand the **Visibility** tab and uncheck the Label, the Units and the Timestamp checkboxes (**leave just the Value** checkbox ticked as in Figure 17).

FF\$01	Text Alignment
	▼ Visibility
	Label
	Units
	Timestamp
	✓ Value

Figure 17. Visibility tab of the Value symbol

- Step 7. Resize the symbol at will and place it at the top of the display as a title.
- Step 8. Copy and paste the text box twice
- **Step 9.** Drag the Product and the Production Line attributes on top of the copied symbols to replace their value (make sure you drag them on top of the existing symbols to replace their content):



Your new title should look as follows:



Step 10. Switch between assets using the Asset Dropdown Menu and verify that the title changes dynamically.

More on String Builder – A glimpse into substitution parameters

Let's look at another example of how to use String Builder attributes. Remember the TagName attribute from Section 2.4.1? Move to the **Production_FFS** template and look at how these TagName attributes are built:

Library	Sandbo	x_FFS					
Sweet & Savory Corp	Genera	Attribute Templates	Ports Ana	alysis Templates	Notif	ication Rule Template	S
							Group by: 🗹
Element Templates	Filter			Q	•	Name:	TagName
The second		i ♦ & Name		De@pt	^	Description:	
Production_Packaging Imm G Sandbox_FFS	□ (Category: Identificat	tion			Properties:	Hidden
G Sandbox_Packaging		🖳 Asset				Categories:	
		📇 Last Servi	ce Date			Default UOM:	<none></none>
🗄 📸 Transfer Templates		🖳 Model				Value Type:	String
Model List		S Nominal M	ax Throughpu	ıt		Default Value:	Press F2 to show the Text Visualize
🔅 🗝 🕏 Reference Types	⊢					Display Digits:	-5
🗄 🔤 Tables		E Product				Data Dafarana	Otrine Duildes
Model Specifications		E Production	n Line			Data Reference:	String builder
III Product Assignment	□ (Category: Overall Eq	uipment Effec	tiveness			Settings
Production_FSS_Maintenance Sandbox_FSS_Maintenance		🔶 🏾 🏹 Availability	/			"%Element%";".";	"% Attribute%";
Table Connections		🧏 📑 TagNam	e 🗲				
E Categories		L					

"%Element%";".";"%..|Attribute%";

This syntax is used to concatenate:

- (1) the name of the AF element that the attribute resides on (%Element%)
- (2) the dot character (".")
- (3) the name of the parent attribute with respect to the TagName attribute (%.. | Attribute%)

Steps (1) and (3) use so-called **substitution parameters** (%Element% and %.. |Attribute%), which minimize the need for manual configurations. String Builder can leverage <u>substitution parameters</u>, as well as <u>attribute referencing</u> and <u>functions</u>. If you want to learn more, we suggest you visit:

- <u>String Builder data references</u>
- <u>Substitution parameters in data references</u>



2.4.7. OPTIONAL - Categories

Asset Framework allows you to organize objects into categories, which you can define yourself. Their purpose is to help you find objects more easily. Objects can belong to multiple categories.

Each object type has its own categories. You cannot apply categories from one object type to an object of another type (e.g. you cannot apply an element category to a table). AF supports the following category types:

- Analysis
- Attribute
- Element
- Notification Rule
- Reference Type
- Table

When you search for an object in PI System Explorer, you can use the category as a filter to reduce the list of results. Categories are useful in many other situations, for example when:

- Selecting attributes in PI Vision
- configuring a Rollup analysis to pick up all attributes belonging to a given category
- filtering the analyses that you want to backfill / recalculate
- filtering AF objects when using the PI Integrator for Business Analytics or PI OLEDB Enterprise
- organizing and maintaining your Asset Framework server

In our sandbox environment, the AF attributes have already been categorized into:

- Identification: static metadata providing contextual information about the asset
- **Overall Equipment Effectiveness**: attributes involved in the calculation of the OEE
- **Process Data**: all real-time data coming from the production line

The maintenance engineer noticed that a couple of attributes under the Process Data category could be better classified. He wants to create a dedicated category for the **Shift** and the **Operator** attributes and call this category **Shift Information**. Let's help him out!

Step 1. From the Library tab, select template Sandbox_FFS \rightarrow Attribute Templates tab.

Step 2. Select both the Operator and the Shift attributes holding down the *Ctrl* keystroke.





Step 4. Deselect Process Data and click on New Category... :

Categorize		×
Use the checkboxes to assign categories:		
Name Identification Overall Equipment Effectiveness Process Data	Description	
New Category	ОК	Cancel
Step 5. Name the Category Shift In	<i>formation</i> and hit	💐 Check In

🚰 Attribute	Category Properties			×
General				
Name:	Shift Information			
Description:	Shift Information			
Type:	Attribute Category			
	Security			
	OK Cancel Apply	Check	In	

Step 6. Click OK again to close the Categorize pop up window.

You should see that the two chosen attributes now appear under the Shift Information category:



You can explore the existing Categories in the Library by navigating to Categories \rightarrow Attribute Categories. This can be useful when renaming or deleting Categories.

Library	Attribute Categories
 Sweet & Savory Corp Templates Templates Element Templates Model Templates Transfer Templates Transfer Templates Tables Table Connections Categories Analysis Categories Attribute Categories Reference Type Categories Table Categories Reference Type Categories Table Categories 	Filter Name Shift Information Overall Equipment Effectiveness Process Data
Step 7. Important: Click on	to save your changes.

Step 8. Move to PI Vision, refresh the page and switch to Edit mode (button in the top-right corner).

Step 9. Navigate to one of the FFS units under the Sandbox environment and verify that the new categorization has been applied. Figure 18 shows that Shift and Operator now appear under the Shift Information category.



Figure 18. PI Vision displays AF attributes by category

\checkmark	Use Categories to organize, manage and filter AF objects such as attributes, analyses and elements. This will help you search for what you need when building
	reports or displays with any PI Client tool such as PI Vision, PI DataLink and PI
Best Practice	ProcessBook.

3. Working with Asset Analytics

This Chapter focuses on some important aspects of Asset Analytics. Section 3.1 introduces what Asset Analytics can do for your business, while Section 3.2 guides you through various scenarios where analyses can be improved in terms of performance and manageability.

3.1. Why Asset Analytics

What is the overall efficiency of a process? Which facility had the highest average production last week? And what about last month?

Users often run calculations on raw data to make key decisions such as which assets require maintenance, or to determine the profitability of a production site. However, these calculations often reside in spreadsheets, with data ranges, methods, or errors as numerous as the individuals who maintain them.

Asking one question can therefore lead to a range of different, and potentially contradictory, answers.

With Asset Analytics, you can easily configure server-side calculations with pre-built functions and *Intellisense* that will auto-populate suggestions as you type. Expressions can be basic, one-line calculations like the standard deviation of a voltage measurement, or multi-step calculations to determine overall equipment efficiency. Asset Analytics also provides a simple configuration option to write calculation results back to PI Points on either a timed (periodic) or events-driven schedule. All of this is possible with a no-coding-required user interface and enables large-scale deployment of calculations that are more manageable and powerful than could be handled within a typical spreadsheet.

For those using Performance Equations on the Data Archive, the same functionalities (and more!) are available in Asset Analytics where analyses can be easily deployed, backfilled, and managed at scale from a single user interface. (2)

Assets Analytics has become the calculation engine of choice for the PI System thanks to its ease of use and vast capabilities. With Assets Analytics, you can take advantage of your AF hierarchy and perform calculations on all your assets.

However, an overloaded **PI Analysis Service** (the Windows service running Asset Analytics) will result in lagging calculations, skipped evaluations, and slow backfilling/recalculations. This chapter provides best practices that will help you achieve optimum scalability and performance. For more information, please visit page: <u>PI Analysis Service Best Practices</u> (5), which also provides considerations on the PI Analysis

Service global configuration parameters and the Data Archive tuning parameters relevant to the PI Analysis Service.

3.2. Asset Analytics Best Practices - Directed activities

This Section presents some best practices applied to our case study. This time we don't start from PI Vision, as some of the potential issues we'll tackle might not have immediate repercussions on the enduser experience. However, we provide you other ways of assessing the impact of your configuration choices.

3.2.1. Use of Expression Variables

The chief process engineer wants to see in a glance whether the **hourly average** of the **Packing Rate per Minute** is meeting the target **Packing Target per Minute**.

The analysis should output one of the following messages:

- Far Behind Target Rate
- Behind Target Rate
- On Target
- Ahead of Target Rate
- Far Ahead of Target Rate

For this purpose, the analysis **SB_PackingRateStatus** has already been configured in the Sandbox environment.

Go to Sandbox \rightarrow Packaging \rightarrow FFS01, select the Analyses tab and check the configuration of the **SB_PackingRateStatus** analysis. You can click on the Expression field to expand it and view the analysis logic:

Elements	FFS01
Elements 🗇 PI Data Archive 🎯 Production	General Child Elements Attributes Ports Analyses Notification Rules Version Image: Second Se
Sandbox Packaging Packaging FF501 FF502 FF503 FF504 FF505 FF506 FF506	Image: Constraint of the sector of the se
	● f⊗ Service Time Calc Add a new variable
	Name Expression
	Variable2 if ((TagAvg('Packing Rate Per Minute','*-1h','*') -
Elements Event Frames	<pre>if ((TagAvg('Packing Rate Per Minute','*-1h','*') - 'Pa 'Packing Target per Minute') > -10 and (TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per Minu Rate" Else If ((TagAvg('Packing Rate Per Minute','*-1h'))</pre>

The full logic is reported below:

if ((TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per Minute') < -10) then " Far Behind Target Rate" Else if ((TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per Minute') > -10 and (TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per Minute') < 0) then "Behind Target Rate" else if ((TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per Minute') > 0 and (TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per Minute') > 0 and (TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per Minute') > 0 and (TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per Minute') < 10) then "Ahead of Target Rate" Else If ((TagAvg('Packing Rate Per Minute','*-1h','*') - 'Packing Target per Minute') > 10) then "Far Ahead of Target Rate" Else "On Target"

The expression looks rather lengthy and complex, but in fact it can be reduced to blocks of four steps:

- calculate the hourly time-weighted average of the Packing Rate per Minute using the TagAvg() function
- 2) perform a subtraction between the hourly average and the Packing Target per Minute
- 3) perform a logic operation (< or >)
- 4) if the condition at point 3 evaluates to True, write a String value to the PI Point data reference attribute Packing Rate Status
Imagine that the hourly average of **Packing Rate per Minute** falls within the last condition of the IF statement ("*on target*"). How many timeswould we need to perform **steps 1 to 3** before deciding whether to write the result or to proceed to the next condition in the IF statement? [*4 times*!]

A much better option is to use **expression variables** to perform the calculation.

Look at the same analysis in one of the Elements under the **Production environment** (for example **FFS07**). The result of the calculation TagAvg('**Packing Rate per Minute**','*-1h', '*') is assigned to the variable **vAverageBPM**. In practice, the result is cached for the time neededto perform the evaluation. The analysis will only need to evaluate the hourly average once, and then perform steps 3 and 4 above. For more details on how the data cache of the PI Analysis Service works, please refer to <u>The Analysis Service Data</u> <u>Cache (7)</u>.



	Use the little "v" in front of your variables, so you can easily distinguish expression
Ţ	variables versus AF attributes. Also, notice that when you recall a variable name
Tip	you should not use single quotes.

You might be wondering... does it really make a difference in terms of performance of the PI Analysis Service?

In order to test whether this configuration choice makes any difference:

Step 1. Move to element **FFS01** \rightarrow Analyses tab.

Step 2. Right-click on the analysis SB_Packing Rate Status and select Preview Results.

Elements	FFS01									
🖶 Elements	Genera	al C	hild El	ements	Attributes	Ports	Analyse	s Notific	ation Rule:	version
I PI Data Archive		►								
🚊 🗝 Sandbox	0	T	0		Name			Backfillin	a	
🖃 🗇 Packaging				f⊗	Availability	Nonthly	Avera	0		
🗇 FF502	0	T		f⊗	OEE			0		
🗇 FF503				f⊗	OEE - Availa	ability		0		
	0	T		f⊗	OEE - Perfo	rmance		Ø		
🗇 FF506				f⊗	OEE - Quali	ty		Ø		
武 Element Searches	0	T		f⊗	SB_Packing	Rate Sta	atus	- Ø		ew
	0			f⊗	Service Tim	e Calc			XD	elete
									C P	review Results
									K B	ackfill/Recalculate

Step 3. Select the desired time range (e.g. Start Time = t, End Time = *)

Step 4. Click on Generate Results on the top right of the window.

Preview results for SB_Pac	king Rate Status	57		×
Start Time: t	End Time: *		Generate	Results
Trigger Time Variable	Generates results of the analysis given the	e start time and e	nd time, ua	ation

The statistic that we are most interested in is the **Average** evaluation time.

In the example below, we see that the analysis in the Sandbox environment takes on average 1.6 milliseconds to be evaluated (Figure 19), while the analysis under the Production environment takes on average only 0.6 milliseconds (Figure 20). We reduce by over 62% the evaluation time just by using the expression variables!



Figure 19. Preview Results for the analysis "SB_Packaging Rate Status" in the Sandbox environment



Figure 20. Preview Results for the analysis "Packaging Rate Status" in the Production environment

A similar example is available in the PI Square post <u>Asset Analytics Best Practices - Part 1: Use Variables</u> (4). Commenting on the example provided on the post, our AF & Analytics specialist states:

«The analysis with the repeating function took ~6 times longer to evaluate in my system. You may be telling yourself, even though it's slower, it's still only takes about 34ms to complete which isn't much and this isn't a problem.

Best Practice

However, what's extremely important here is to **think about scalability!** If the *[ill-configured]* analysis was based on an analysis template and there were hundreds or thousands of other analyses based on the same analysis template, the performance difference at the group level (all analyses based on the same template) would be **on the order of seconds, perhaps even minutes**. **»**

NERD ALERT - Additional consideration: client vs server evaluations

In our example we used the **Preview Results** tab, which gives us evaluation statistics for **client-side evaluations**. In other words, when you click on Preview Results you are **not** calling the PI Analysis Service itself; instead, you are using the **resources of the client machine** you are on. In our case, we are working in PI System Explorer on the PI Server machine itself, but normally you would open it from your client machine.

The evaluation statistics provided by the Preview Results window are often a bit *pessimistic*, as the machine hosting the PI Analysis Service is usually more performing (in terms of memory and processor speed) than the client machine you use. As a rule of thumb, we suggest making sure that the average (client-side) evaluation time is significantly below the triggering frequency (see next section 3.2.2 for more on analysis scheduling).

Use expression variables whenever you want to repeat a block of your analysis logic, such as a summary operation for the same tag and time span. This will improve the performance of the PI Analysis Service as well as simplify the analysis management.

3.2.2. Analysis Scheduling

The chief maintenance engineer wants to keep track of the **Availability Monthly Average** for the purpose of identifying which FFS units have undergone the longest monthly downtime.

He needs the results of this calculation twice a week for his meetings with the maintenance team.

Let's move to FFS01 and look at the analysis **Availability Monthly Average**. This analysis uses the function **TagAvg()** and spans over the last month of data. What's important to notice here is the **Scheduling** configuration: the analysis is set to trigger on a periodic basis with a **1-minute frequency**!

Elements	F\$01	
🖶 Elements	General Child Elements Attributes Ports Analyses Notification Rules Version	
PI Data Archive	Name: AvailabilityMonthly Avera-	ge
	BEB Description	
Bandbox	Backfilling Ame Backfilling	
🗇 FF501	🕜 🗉 ftØ AvailabilityMonthly Average 🕜 Categories:	~
🗇 FF502	🔗 🗉 f 😣 OEE 🔗 🛛 🖉 Australia Tanana 💿 Expression 🔘 Ro	ollup
🗇 FF503	📀 🔳 🎋 OEE - Availability 📀 🚺 Analysis Type: Event Frame Generati	tion 🔘 SQC
FF504 FF505	🧭 🗉 f 🔅 OEE - Performance 🔗	
🕰 Element Searches		
	Add a new variable	Evaluate
	Name Expression	Output Attribute
	vAvailAvg TagAvg('Availability','*-1mo','*')	Availability Monthly Average
		+ 4
Elements		
Event Frames		
🏭 Library		
🚥 Unit of Measure		
A Contacts	Scheduling: Uvent-Inggerea Penoaic Advanced	
💥 Management	Period: 00h 01m 00s Configure	Connected to the PI Analysis Service.

Why should we trigger the analysis every minute if the results are needed only twice a week?

These 1-minute evaluations pose an unnecessary workload on the PI Analysis Service. The impact becomes even more significant when backfilling or recalculating the analysis over a long period of time, as the PI Analysis Service will have to perform many evaluations "in bulk". For example, given a 1-minute frequency, backfilling over the last 6 months would yield 259 200 evaluations. If we set it to daily frequency, we go down to 180 evaluations for the same time span (less than 0.07% of the initial number of evaluations).

In order to change the scheduling of the analysis:

- **Step 1.** Move to Library \rightarrow Templates \rightarrow Element Templates \rightarrow **Sandbox_FFS.**
- Step 2. Open the Analysis Templates tab.

Step 6. Once you are done, click OK and

- Step 3. Click on the analysis named Availability Monthly Average (Figure 21).
- Step 4. Click on the Configure button and choose Daily (Figure 22).
- Step 5. Under the same tab, we can also specify the time of day we want to run the analysis at. Let's set it to 1 AM (unless otherwise required, it is good practice to set offsets compared to the default time of 12 AM, so that not all daily calculations get triggered at the same time).

Library Sandbox_FFS 🗳 Sweet & Savory Corp General Attribute Templates Ports Analysis Templates Notifica Templates 🚊 🗝 🚮 Element Templates ------ 🔂 Environment 0 • Name 🗄 🗝 🚮 Production_FFS f AvailabilityMonthly Average --- 🔂 Production_Packaging 🗄 🗝 🚰 Sandbox_FFS f⊗ OEE 🦾 🖓 Sandbox_Packaging f⊗ OEE - Availability Event Frame Templates f(x) OFF - Performance 😤 Model Templates ÷.... 🗄 🗝 🃸 Transfer Templates Example Element: Sandbox\Packaging\FFS01 Chumeration Sets 😟 🗠 🔄 Reference Types Add a new variable --- 📺 Tables Ė Table Connections Name Expression --- 🗋 Categories vAvailAvg TagAvg('Availability','*-1mo','*') ----- 🙆 Analysis Categories -- 回 Attribute Categories ····· 🔊 Element Categories ····· 🖻 Notification Rule Categories ----- 🕒 Reference Type Categories 🛄 Table Categories Elements Event Frames 🏭 Library 🚥 Unit of Measure Scheduling: O Event-Triggered Periodic Advanced 2 Contacts Period: 00h 01m 00s Configure 💥 Management

Figure 21. Availability Monthly Average configuration

🛃 Check In



Figure 22. Periodic Schedule tab

If you would like to learn more about the impact of data density and scheduling on the PI Analysis Service, please visit these PI Square posts: <u>Asset Analytics Best Practices - Part 2</u>: <u>Data Density and Data Pattern</u> (9) and <u>Asset Analytics Best Practices - Part 5</u>: <u>Scheduling</u> (10).

NERD ALERT – Exit() function and data density

The analysis could have been configured to skip the evaluation if the current day is not the day when the results are needed. For example, the logic could be:

```
if Weekday('*') <> 2 or Weekday('*') <> 6 then Exit() else TagAvg('Availability', '*-
1mo', '*')
```

NB: The **Exit()** function has been introduced in release **2018 SP2** and is particularly useful when performing high-workload calculations such as summary calculations over extensive periods of time.

Finally, if the input data is very dense, you could split the average calculation in the following way:

- Calculate the daily average (and save the output result in a PI Point). This analysis could be scheduled as Event-Triggered.
- Calculate the monthly average using, as input, the output of the daily calculation. This analysis could be scheduled as Periodic and run daily.

This will reduce the total amount of data used for the monthly calculation by a factor 30!



Don't run an analysis more frequently than it needs to be run, as needless evaluations may pose a significant load on the PI Analysis Service.

3.2.3. Rollup Analyses

A rollup analysis calculates statistics such as sum and average of selected attributes associated with an element.

The chief process engineer needs to monitor the **Average Availability** of the FFS units at the Production Line level. One of his colleagues suggested that he uses a Formula data reference attribute, on the basis that it's easy to set up and it evaluates only *on the fly*, when the result is requested. In order to create a Formula, he assigns each input to a parameter and then writes the equation as in Figure 23.

Click on the **Settings** button to display the parameters and equation in a more user-friendly way (Figure 24):

Elements	Packaging			
Elements	General Child Elements Attributes Ports Analyses	Noti	fication Rules	Version
Data Archive				Group by: 🗹 Category 🗌 Tem
Sandbox	Filter 🔎 🗖	·	Name:	Average Availability
err Packaging	: ■ ♦ 🞗 Name	-	Description:	
🗇 FFS01 🎒 FFS02	Category: <none></none>		Properties: Categories: Default UOM:	<none></none>
FFS03	Average Availability			
	🖩 🔶 🍼 Average Availability - Rollup			percent
			Value Type:	Double
Element Searches			Value:	62.297 %
		Þ	Display Digits:	-5
			Data Referenc	e: Formula
D Elements				Settings
Event Frames				didu a inceasir didu a inceasir didu a inceasir didu a
🎬 Library			A=.VFS01 Av	vailability;B=. \FFS02 Availability;C=. \FFS03 Availability;D=. \FFS04 Availability;E=. bility;F=. \FFS06 Availability;[(A+B+C+D+E+F)/6]

Figure 23. Average Availability calculation – Formula data reference

Parameters		Equations		
A=.,FF501 Availability B=.,FF502 Availability C=., FF503 Availability D=., FF504 Availability E=.,FF505 Availability F=.,FF506 Availability		(A+B+C+D+E+F)/t	5	× × × *
Default Values Allowed tesult JOM: %	Minimum:	Maximum:	Stepped	

Figure 24. Average Availability calculation – Formula settings tab

After implementing the Formula, our chief production engineer finds out an alternative way of configuring this analysis, i.e. using a Rollup analysis within Asset Analytics (Figure 25). The element Packaging uses the AF element template **Sandbox_Packaging**.

Library	Sandbox_Packaging			
Sweet & Savory Corp	General Attribute Templates Ports Analysis Templates Notifi	cation Rule Templa	ates	
Element Templates	😨 📾 Name	Name: Description:	Average Availability - Rollup	^
Production_FFS Production_Packaging Productin_Packaging Production_Packaging Production_Packaging	Image: Average Availability - Rollup Image: Average Availability	Expression Rollup Cevent Frame Generation alyses when created from templat	_ soc	
	Rollup attributes from Child elements of Packaging This element - Packagin To select attributes set criteria below Attribute Name: availability Attribute Level: Root Level Attribute Category:	Sample Child Element V Availability Asset Availability Month V Bags Today Last Service Date Model Moisture Content	 PFS01 Parent Element FFS01 	
Elements Event Frames	Function Output(s) Value At Eval Value At Las	t	OEE Operator	FFS01 FFS01 FFS01
	Average Average Availab		Show more attributes (Showing 22 of total 22 at
Contacts Management	Scheduling: O Event-Triggered O Periodic Advance Period: 00h 05m 00s Configure	ed		

Figure 25. Average Availability – Rollup analysis

The chief production engineer tests both analyses in the Sandbox environment and it looks like they provide the same results down to the first decimal:

Elements	Packaging	9						
Elements Data Archive Production Sandbox Preckaging FFS01	General	Child Eler	nents Name	Attributes	Ports	Analyses	Notification Rule	s V
🗇 FFS02	8 🖻							
🗇 FFS03 🎁 FFS04			🗉 Ave	erage Availa	ability		62.297	%
🗇 FFS05 🎯 FFS06	0	•	Ø Ave	erage Availa	ability - R	tollup	62.292	%

Figure 26. Average availability - comparison between Formula data reference and Rollup analysis

Which of the two approaches would you suggest using and why?

See what happens when you delete a couple of Elements from the Sandbox environment.

- 1. Move to the Elements tab and select **FFS01**.
- 2. Right-click and select **Delete...** In the opening pop-up windows choose:

Permanently delete; this action is irreversible

3. Now hit Check In and Refresh

You will notice that the rollup analysis that uses a Formula data reference shows error message:

Unknown Attribute '.\FFS01|Availability' in configuration of formula in attribute 'Sandbox\Packaging|Average Availability'.

Elements	Packaging							
Elements Data Archive Production	General Child E	ements Attributes Ports Anal	vses Notification Rules Version					
⊡	Image: Second							
🗇 FFS04	0 E	💷 Average Availability	Unknown Attribute '. \FFS01 Availability' in configuration of formula in attribute 'Sandbox\Packagi					
☐ FFS06	ø 🗉 🔶	🎺 Average Availability - Rollup	62.372 %					
······ 🔍 Element Searches								

Figure 27. Error message following the deletion of FFS01.

The error message pops us because the Formula data reference is not "smart" and does not update whenever a Child Element of the Packaging element is added or removed. Moreover, the "FFS count" at the denominator of the Formula expression (6, i.e. the number of FFS units) must be manually modified if the unit count changes.

On the contrary, a *proper* Rollup analysis built with Asset Analytics identifies its input attributes each time it is executed, and automatically includes any new attributes that meet its selection criteria (or removes inputs that are no longer available).

If you haven't had a chance to work with Rollup analyses, take a few minutes to examine the configuration of the analysis **Average Availability – Rollup**.

You will notice that you can apply the rollup to Child Elements or to attributes of the Element itself.

You can also apply filters such as categories and elementtemplates (for example, you could create a Rollup analysis at the Production level that considers only the Elements built on the template **Sugar Production_FFS**).

Which scenarios in your industry can you think of where rollup analyses could be used?

	You can add or delete elementsor attributes in your hierarchy without the need to
V	update Rollup analyses. Because a Rollup identifies input attributes each time it is
Best Practice	executed, it automatically includes any new attributes that meet its selection
	criteria.

The following two sections are marked as *nerd alert*. Feel free to go through them should you still have time, or save them for when you are back home with your PI World gadgets!

3.2.4. NERD ALERT – Analysis Organization and Consolidation

Organizing analyses in a clear manner is not just about the aesthetics.

Take for example the OEE analysis configured in template **Sandbox_FFS** and in template **Production_FFS**.

brary	Sandbox_FFS	
Sweet & Savory Corp → Templates → Element Templates → Environment → Production_FFS → Production_PFS	General Attribute Templates Ports Analysis Templates	Notification Rule Templates
		Name: OEE
	💿 🖪 Name	Description:
	f⊗ AvailabilityMonthly Average	Categories: Overall Equipment Effectiveness
	f⊗ OEE	Expression Rollup
🔂 Sandbox_Packaging	f⊗ OEE - Availability	Analysis Type:
Event Frame Templates	f⊗ OEE - Performance	Enable analyses when created from template
Hodel Templates	fix OEE - Quality	
- 🔞 Enumeration Sets	f⊗ SB Packing Rate Status	•
- 🔄 Reference Types		
	Example Element: Sandbox\Packaging\FFS01	
- Categories	Add a new variable	Evaluate
Analysis Categories	Name Expression	Output Attribute
Attribute Categories		oupernande
🖻 Notification Rule Categories	OEE 'Availability'*'Quality'*'Perfo	ormance'/10000
🔄 Reference Type Categories		

Figure 28. Sandbox_FFS

Library	Production_FFS					
Sweet & Savory Corp	General Attribute	Templates Ports Analysis Templates	Notification Rule Ter	nplates OEE		
Comparison of the second	Image: Second state Image: Second state </td <td>e lability Monthly Average ing Rate Status ice Time Calc : <u>Production\Salt Packaging\FFS09</u></td> <td colspan="2">Description: Categories: Overall Equipment Eff © Expression Analysis Type: Cevent Frame Gene Cevent Fr</td> <td colspan="2">ectiveness) Rollup eration O SQC m template</td>	e lability Monthly Average ing Rate Status ice Time Calc : <u>Production\Salt Packaging\FFS09</u>	Description: Categories: Overall Equipment Eff © Expression Analysis Type: Cevent Frame Gene Cevent Fr		ectiveness) Rollup eration O SQC m template	
Im Table Connections	Add a new varia	ble			Eval	uate
Analysis Categories	Name	Expression			Output Attribute	
🗃 Element Categories 🖻 Notification Rule Categories	Availability	TimeEq('Running State','t','	*',1)/DaySec('	*')*100	Availability	۲
Reference Type Categories Table Categories	Quality	Rand(80,20)			Quality	8
	Performance	'Packing Rate Per Minute'/'Packing Target per Minute'*100			Performance	⊗ '
a Flements	OEE	'Availability'*'Quality'*'Pe	rformance'/100	90	OEE	۲

Figure 29. Production_FFS

The OEE calculation in the Sandbox environment is split up into four distinct analyses, while in the Production environment there is only one analysis comprised of four variables.

You can choose between consolidating and splitting analysis expressions by considering the three criteria below:

• **Dependency**. If one analysis output depends on the results of another analysis, it often makes more sense to consolidate the analyses. This ensures that the dependent expression gets

evaluated only when all the inputs have been evaluated using the most recent data. In our example, the OEE should only be calculated when Availability, Quality and Performance have been calculated.

• Scheduling. If a group of related expressions should be evaluated at the same time, it is usually best to consolidate them.

However, if some of these expressions need to be evaluated at a much higher frequency, it is better to create independent analyses to avoid needless evaluations of the slow-paced expressions.

 Manageability. Consolidating analyses facilitates management and troubleshooting. In case the analysis is failing, using the Evaluate or Preview Results options will guide you to the faulty expression.



3.2.5. NERD ALERT - Analyses in Error

By now you'll have noticed that the **Service Time Calc** analyses are all in error state (Figure 30).

The chief maintenance engineer had planned to use these analyses to calculate the number of days since the last maintenance service. However, other tasks took priority and the analyses were left in a halfcooked state.

Elements	FFS03
Elements DI Data Archive Comparison Production	General Child Elements Attributes Ports Analyses Notification Rules Version Image: Second Sec
	● ■ Name Backfilling ⑦ ■ f(k) Availability/Monthly Average
	jv jv occ jv fv OEE - Availability jv fv OEE - Performance
i ☐ FFS06 Element Searches	Image: Status Image: Status Image: Status Image: Status Image: Status Image: Status
Errors And Warnings No output is defined.	

Figure 30. Service Time Calc analysis in error state

Why is this relevant for us?

If analyses are enabled before they are configured correctly, the PI Analysis Service will automatically

try to restart them every 15 minutes, causing unnecessary strain.

In order to disable all the analyses in error state, proceed as follows:

- 1. Go to the Management tab.
- 2. Click on the symbol to add a new search (Figure 31)



Figure 31. Add new search under Management tab

- 3. Name your search Analyses in Error state (Figure 32)
- 4. Add the filter criterion **Service Status** (Figure 32)

Search Name:	Analyses in Error sta	ate ←				
Add criteria to search for specific items.						
Add Criteria 🔻]	_				
Name		eria will be displayed.				
Descript	tion					
Element	t Name					
Templat	e					
Categor	у					
Enabled	/Disabled					
Service	Status 🗲 🗕					
		_				
		OK	Cancel			

Figure 32. Search configuration

5. Select the Error status and then click OK (Figure 32)

Search Name: An	alyses in Error state	
Service Status:	Running ~	×
	Running	
Add Criteria 🔻	Stopped	
* 4	Warning	
** Analyses that mat	Error	
	Suspended ¹	
	OK Cancel	

- 6. Select all the filtered analyses using the first checkbox on the top (Figure 33)
- 7. Disable the selected analyses (Figure 33)

		Analyses								
		20 t	tal anal	yses s	elect	ted (20 on this page)			1 - 20 of 20 🛛 <	Operations
1			Status	0		Element	Name	Template	Backfilling	Enable Disable selected analyses
<u>.</u>		-	0		f09	Production\Salt Packaging\FFS07	Packaged Weight			Enable Disable automatic recalculation for selected analyses
		1	0		f69	Sandbox\Packaging\FFS03	Service Time Calc	Service Time Calc		
		-	0		f6)	Production\Salt Packaging\FFS07	Service Time Calc	Service Time Calc		analyses
	-	-	0		f(>)	Production\Salt Packaging\FFS17	Service Time Calc	Service Time Calc		
	-	-	0		f(*)	Production\Sugar Packaging\FFS20	Service Time Calc	Service Time Calc		
		•	0		f(\$)	Production\Sugar Packaging\FFS16	Service Time Calc	Service Time Calc		
	•	-	0		f(\$)	Production\Salt Packaging\FFS19	Service Time Calc	Service Time Calc		
		•	0		f60	Production\Sugar Packaging\FFS15	Service Time Calc	Service Time Calc		
		-	0		f69	Production\Salt Packaging\FFS18	Service Time Calc	Service Time Calc		
		1	0		f6)	Production\Sugar Packaging\FFS14	Service Time Calc	Service Time Calc		
		-	0		f6)	Production\Sugar Packaging\FFS13	Service Time Calc	Service Time Calc		
		-	0		f(*)	Production\Salt Packaging\FFS12	Service Time Calc	Service Time Calc		11
		-	0		f(x)	Production\Salt Packaging\FFS11	Service Time Calc	Service Time Calc		
		1	0		f(\$)	Production\Sugar Packaging\FFS10	Service Time Calc	Service Time Calc		Pending Operations
		•	0		f⊗	Production\Salt Packaging\FFS09	Service Time Calc	Service Time Calc		No pending operations
		-	0		f60	Production\Salt Packaging\FFS08	Service Time Calc	Service Time Calc		
		1	0		f69	Sandbox\Packaging\FFS04	Service Time Calc	Service Time Calc		
		-	0		f69	Sandbox\Packaging\FFS06	Service Time Calc	Service Time Calc		
		-	0		f60	Sandbox\Packaging\FFS01	Service Time Calc	Service Time Calc		
	-	~	0		f(*)	Sandbox\Packaging\FFS02	Service Time Calc	Service Time Calc		

Figure 33. Select and disable the analyses in error

The analyses should now display as disabled (Figure 34).

Ana	lyses							
20 t	otal analy	ses s	select	ted (20 on this page)			1 - 20 of 20 < >	Operations
~	Status	0	A	Element	Name	Template	Backfilling	Enable Disable selected analyses
1	0		f60	Production\Salt Packaging\FFS07	Packaged Weight			Enable Disable automatic recalculation for selected analyses
1	\oslash		f(x)	Sandbox\Packaging\FFS03	Service Time Calc	Service Time Calc		
1	0		f60	Production\Salt Packaging\FFS07	Service Time Calc	Service Time Calc		Queue Cancel backfilling or recalculation for selected analyses
1	0		f(s)	Production\Salt Packaging\FFS17	Service Time Calc	Service Time Calc		
1	\oslash		f(\$)	Production\Sugar Packaging\FFS20	Service Time Calc	Service Time Calc		
1	0		f60	Production\Sugar Packaging\FFS16	Service Time Calc	Service Time Calc		
1	\oslash		f60	Production\Salt Packaging\FFS19	Service Time Calc	Service Time Calc		
1	\otimes		f69	Production\Sugar Packaging\FFS15	Service Time Calc	Service Time Calc		
1	0		f60	Production\Salt Packaging\FFS18	Service Time Calc	Service Time Calc		
1	\otimes		f(\$)	Production\Sugar Packaging\FFS14	Service Time Calc	Service Time Calc		
1	0		f60	Production\Sugar Packaging\FFS13	Service Time Calc	Service Time Calc		
1	\oslash		f⊗	Production\Salt Packaging\FFS12	Service Time Calc	Service Time Calc		
1	\oslash		f60	Production\Salt Packaging\FFS11	Service Time Calc	Service Time Calc		
1	\oslash		f60	Production\Sugar Packaging\FFS10	Service Time Calc	Service Time Calc		Pending Operations
1	\oslash		f69	Production\Salt Packaging\FFS09	Service Time Calc	Service Time Calc		Disabling 20 analyses Dismiss
1	\otimes		f60	Production\Salt Packaging\FFS08	Service Time Calc	Service Time Calc		Time Submitted: 12/13/2019 8:51:57 AM
1	0		føð	Sandbox\Packaging\FFS04	Service Time Calc	Service Time Calc		Queued by: PISCHOOL\student01
1	\oslash		f60	Sandbox\Packaging\FFS06	Service Time Calc	Service Time Calc		📀 Complete
1	0		f(x)	Sandbox\Packaging\FFS01	Service Time Calc	Service Time Calc		
1	0		f69	Sandbox\Packaging\FFS02	Service Time Calc	Service Time Calc		

Figure 34. Analyses in Disabled state

Should you want to view all the disabled analyses (for example when cleaning up your database from analyses that are no longer needed), you can use the default search option "Disabled" (Figure 35).

Management	
Choose a type Analyses Notification Rules	
Analysis Searches	
+ 🗙	
All	•
Enabled	•
Disabled	•
Analyses in Error state	

Figure 35. Disabled analyses filter

A detailed example of the impact of analyses in warning or error states on the PI Analysis Service is provided in the PI Square post <u>Asset Analytics Best Practices - Part 4</u>: <u>Analyses in Warning or Error</u> (11).



The work you have done

Remember all the issues we encountered when exploring the PI Vision displays in Section 2.3?

Let's recap what the issues were and how we solved them. Give yourself a big pat on your back for your hard work!

Issue # 1. Asset FFS05 is not available in the PI Vision Asset Dropdown Menu

Solution # 1. FFS05 was not using an AF element template. Once we assigned it a template, the asset became available in PI Vision.

Best Practice #1. Always use AF templates to make PI Vision displays reusable!

Issue # 2. Trends and statistics for FFS units 3,4 and 6 are identical.

Solution # 2. FFS03, FFS04 and FFS06 have an additional attribute (Sugar Grain Size) configured using the "Allow Extensions" option, and a typo was made in the attribute configuration. We used a Derived Template for these FFS units, and this fixed the issue.

Best Practice # 2. Derived Templates, sometimes in combination with the **Excluded** attribute property, is the way to go to ensure consistency in your Asset Framework experience.

Issue # 3. The attribute Sealer temperature shows values outside the expected range.

Solution # 3. The Source Unit of Measure of the Sealer temperature had to be adjusted to reflect the UOM of the raw data stream.

Best Practice # 3. Asset Framework can automatically handle **UOM conversions** for you, make good use of this feature and don't forget to set the Source UOM explicitly.

Issue # 4. The attribute Last Service Date slows down the loading of the PI Vision display.

Solution # 4. The externally linked AF Table used by the Last Service Date attributes had a large number of rows. We parametrized the query and this greatly accelerated the data retrieval time.

Best Practice # 4. Parametrized queries are your friendswhen retrieving data from large tables in external databases.

Issue # 5. Asset FFS04 shows "No Data" for the Nominal Max Throughput.

Solution # 5. The Model attribute was using a static (*<none>*) data reference. We used an Enumeration Set and set each FFS unit's model choosing from a dropdown list.

Best Practice # 5. Enumeration Set help avoiding typos and ensuring consistent nomenclature throughout the database.

Issue # 6. The title of the display Sandbox_FFS_Monitoring_Faulty_Title does not update with the chosen asset.

Solution # 6. The title used in the display was a static text box. We used a String Builder attribute which dynamically points to the name of the AF element.

Best Practice # 6. String Builder data reference and substitution parameters are very powerful tools for handling string values and creating dynamic pointers.

Finally, we saw how **AF Categories** help us organize, manage and filter AF objects such as attributes, analyses and elements.

Concerning **Asset Analytics**, we discussed:

- the usefulness of expression variables. In our example, we used an expression variable to store the result of a TagAvg() expression. Expression variables improve the performance of the PI Analysis Service as well as simplify the analysis logic;
- 2. the importance of setting the **analysis scheduling** having in mind the required update frequency and the workload on the PI Analysis Service;
- the advantage of Rollup calculations available in Asset Analytics compared to other solutions. Rollups automatically detect any modifications to the input attributes at execution time and update their behavior accordingly;
- the choice between consolidating or splitting expressions into analyses depending on Dependency, Scheduling and Manageability;
- 5. the reason why analyses in **Warning or Error state** should be disabled until they are fixed.

Goodbye!

This Lab hopefully provided you with some good tips on how to improve your data infrastructure in Asset Framework and Asset Analytics.

For time constraints, we left out several aspects and features of Asset Analytics, such as Event Frame Generation, backfilling versus recalculation, and the Analysis data reference, to name just a few. We recommend exploring the **References** and the **Additional Resources** Sections below to keep learning and perfect the Noble Art of Asset Framework and Asset Analytics!

We would like to warmly thank you for your participation, and we hope to see you at the next PI World!

References

1. OSIsoft. A Journey from Historian to Infrastructure. [Online] 2015. [Cited: 12 12, 2019.] https://www.osisoft.com/White-Papers/A-Journey-from-Historian-to-Infrastructure/.

2. Top 10 Reasons to Love Your Modern PI Server. [Online] 2016. [Cited: 12 12, 2019.] https://www.osisoft.fr/white-papers/top-10-reasons-to-love-your-modern-pi-server/.

3. Global Sources. [Online] Global Sources. [Cited: 01 31, 2020.] https://www.globalsources.com/gsol/I/Powder-packing/p/sm/1163559821.htm#1163559821.

4. OSisoft. How AFTable Caching Works. [Online] 2019. [Cited: 12 12, 2019.] https://customers.osisoft.com/s/knowledgearticle?knowledgeArticleUrl=KB00539.

5. OSIsoft. Asset Analytics Best Practices - Part 3: Input Attributes. [Online] 4 22, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/04/22/asset-analytics-best-practices-part-3-input-attributes.

6. —. PI Analysis Service Best Practices. [Online] 2019. [Cited: 12 12, 2019.] https://customers.osisoft.com/s/knowledgearticle?knowledgeArticleUrl=KB01641.

7. The Analysis Service Data Cache. [Online] 2 28, 2019. [Cited: 12 12, 2019.] https://customers.osisoft.com/s/knowledgearticle?knowledgeArticleUrl=KB01495.

8. OSIsoft. Asset Analytics Best Practices - Part 1: Use Variables. [Online] 3 21, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/03/21/asset-analytics-best-practices-part-1-use-variables.

9. —. Asset Analytics Best Practices - Part 2: Data Density and Data Pattern. [Online] 03 28, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/03/28/asset-analytics-best-practices-part-2-data-density-and-data-pattern.

10. —. Asset Analytics Best Practices - Part 5: Scheduling. [Online] 7 11, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/07/11/asset-analytics-best-practices-part-5scheduling.

11. —. Asset Analytics Best Practices - Part 4: Analyses in Warning or Error. [Online] 5 29, 2019. [Cited: 12 12, 2019.] https://pisquare.osisoft.com/people/sraposo/blog/2019/05/29/asset-analytics-best-practices-part-4-analyses-in-warning-or-error.

Additional resources

- <u>LiveLibrary</u>
- OSIsoft Learning
- Public class: "Building PI System Assets and Analytics with AF" (course material freely downloadable at the link "Download Course Content" at the bottom of the page).





PI SYSTEM LEARNING MADE EASY!

Accelerate success with the new OSIsoft Learning platform.

VISIT LEARNING.OSISOFT.COM



© Copyright 2020

OSIsoft, LLC