



vCampus **Live!** 2011

Discover How to Programmatically Generate PI Coresight Displays with the Right Data Items and Time Range

Michael Weiss, OSIsoft



Agenda

- What is PI Coresight?
- Integrating with PI Coresight
- Code sample
- PI ProcessBook Add-In sample
- Discussion



What is PI Coresight?

PI Coresight is an intuitive, web-based tool that delivers fast, easy, and secure access to all your PI System data.

With PI Coresight you can easily perform ad hoc analysis, discover answers, and share your insights with others.



PI Coresight Architecture

- **Silverlight User Interface**
- **Web Services in middle tier**
- PI System and SQL Server



PI Coresight Demo

Watch for...

- “Displays”
- “Symbols”
- The URLs in the browser





PLEASE
PAUSE FOR
DEMO



Why Integrate?

Allows a user to quickly open “interesting” data in PI Coresight without requiring a preexisting display or manual work.

- Selected data items in PI Process Book, PI DataLink, PI WebParts, PI Notification (likely future features available from OSIsoft)
- Custom applications



Building a sample application

1. Find and connect to the PI Coresight Web Service
2. Make the Web Service call to create a display
3. Open the Display



Step 1: Find and Connect



Find and connect to service

- We'll need the URLs to both UI website and the Web Services
- Stored in site's default PI AF server by PI Coresight installer
- PI AF Path:
\\AFServer\Configuration\OSIsoft\PI Coresight | **Coresight URL**



Find and connect - Authentication

- Any application that connects to the PI Coresight Web Services must support Windows Authentication.
- By default the authentication mode is “Negotiate” (Kerberos).



Adding a Web Service Reference

- .NET applications require a Web Reference in order to call the Web Service.
- Other platforms may require a similar step.
- `http://<yourwebserver>/Coresight/Services/Storage/CoresightStorage.svc`



Web Reference, continued

Add the **bolded** line temporarily to Coresight\Services\Web.config:

...

```
<serviceBehaviors>
```

```
<behavior name="CoresightBehavior">
```

```
<serviceThrottling .../>
```

```
<serviceMetadata httpGetEnabled="true" />
```

```
</behavior>
```

...





PLEASE
PAUSE FOR
DEMO



Step 2: Make the call



Making the call - NewAdHocDisplay

NewDisplayInfo **NewAdHocDisplay** (
 string *cog*,
 string *namePrefix*,
 AccessLevel *level*)



The COG

- A common way to share data and context between PI System applications
- By data we mean PI Tags and/or PI AF Attributes grouped into an abstract symbols
- By context we mean a time range
- XML
- Focus on the COG features PI Coresight Trends support



Major COG elements

Databases

A list of PI System and PI AF servers

Datasources

PI Tags or AF Elements (links to a Database)

Symbols

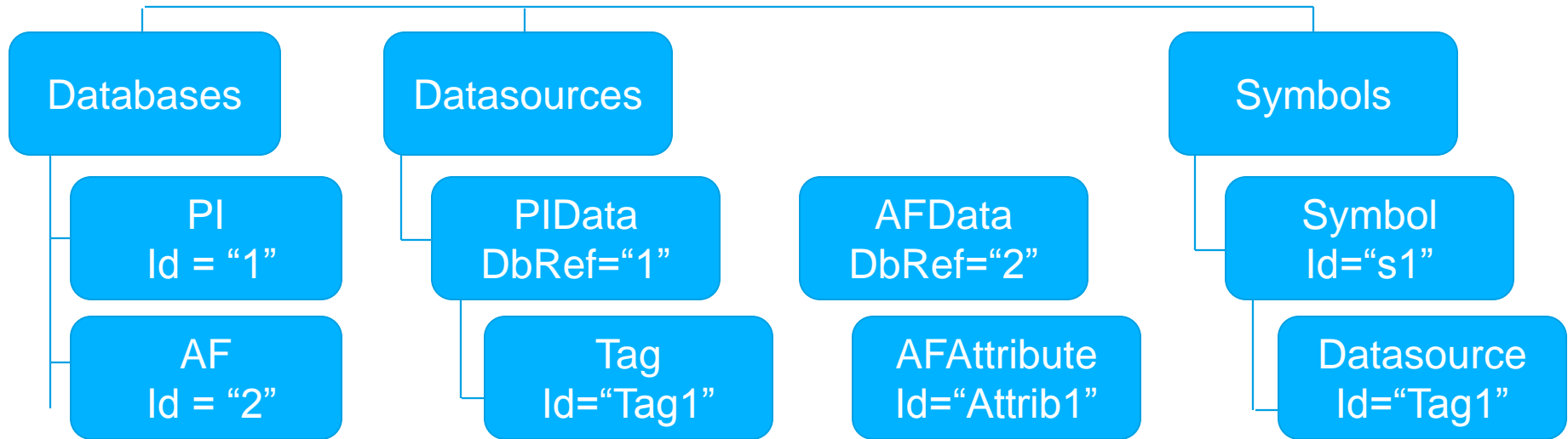
Groups of Datasources (links to one or more Datasources)

Context

Time ranges (links to nothing, for whole Display)



COG Structure



1 per unique PI
Server or PI AF
Database

1 pair per unique Tag or Attribute

1 per Trend
symbol, multiple
children =
multiple Traces



COG Time

- Under an XML Tag: <Context><TimeRange>
- **Date and Time (Rfc3339)**
"yyyy'-'MM'-'dd'T'HH'::mm'::ss'.FFFFFFFFK"
- **Just time**
"HH'::mm'::ss'.FFFFFFFFK"
- Or PI's relative time notation ("*", "*-8h", etc)



The other parameters

Name Prefix

The name of the new Display. PI Coresight will append a unique number to this name.

Access Level

Should the new Display be Public (anyone can **view**) or Private (accessible only by the author).





PLEASE
PAUSE FOR
DEMO



Step 3: Open the Display



Opening the Display

- Web Service call returns new Id
- Create a URL
- Email it





PLEASE
PAUSE FOR
DEMO



Not shown

- Calling Establish
- Localization support
- Client GUIDs
- XSD Validation
- Programmatically creating COGs



PI ProcessBook Add-in





PLEASE
PAUSE FOR
DEMO



What's subject to change?

Endpoints

- The paths of the URLs may change
- AF Configuration may include additional URLs
- Method calls intended for external consumption may be grouped together into a single “endpoint”

Code

- Expect code libraries (if provided) and XSDs will be updated. Service References may need a refresh.



Thank you

