



*ACE 2011 International*

# Understanding Federation and Web Services

---

[www.aras.com](http://www.aras.com)



# Welcome

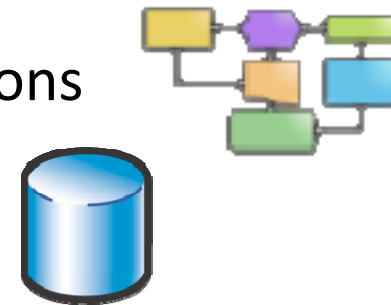


## Session Goals

□ Previous session covered overall Integration strategies and a focus on how to integrate desktop applications (authoring tools)

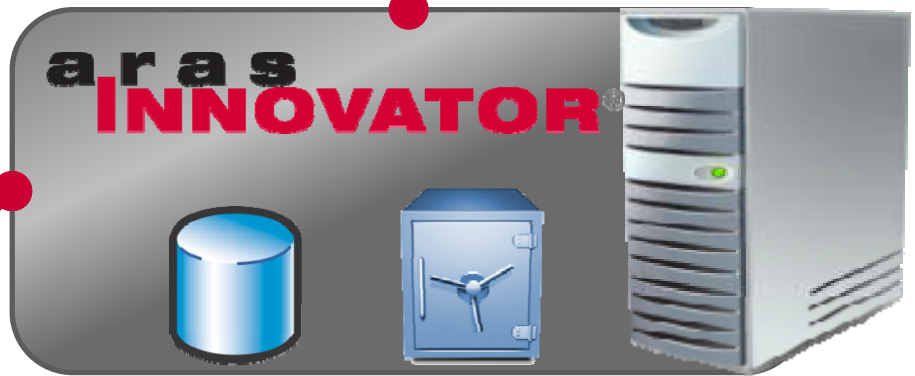
□ This session: **PLM Server ↔ other business servers**

- Creating server peer-to-peer configurations
- Data and Process status exchange
- For the end-users, blurring the lines between systems and presenting all their relevant data in a single context

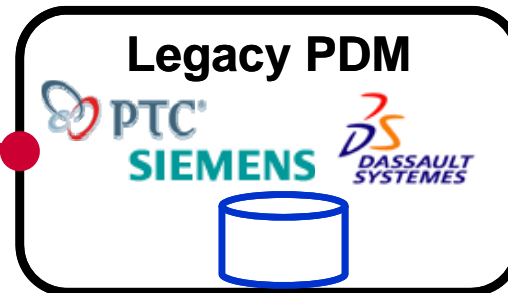


# Session Topic

1. Pull data from external apps into PLM to use in Client, Reports, workflows
2. External apps using PLM as a data source



Database File Vault



# Two Use Cases



**1. Data in the ERP system or your legacy applications is relevant to the PLM users. We want this data and PLM data together in the right context on the PLM user interface -- like a Mash-Up**

- Corporate Document Management office files
- ERP for costing or inventory records
- CRM customer information

**2. Other systems need access to data records or files stored in the PLM system**

- Shop floor viewing of Drawings through MES
- Quality system view of Part-BOM structure

# Two Use Cases



- **NOTE:** for many commercial systems, such as SAP and SharePoint, you can find a commercial 3<sup>rd</sup> party connector from an Aras partner.
  - For this discussion, it is more interesting to imagine that we are building an integration from scratch.
- **NOTE:** it's always possible to batch transfer data from one system to another. We (IT folks) have been doing that forever.
  - For this discussion it is more interesting to talk about how we build real-time connectors

# Two Use Cases



- 1. Data in the ERP system or your legacy applications is relevant to the PLM users. We want this data and PLM data mixed together in the right context on the PLM user interface -- like a Mash-Up**

## Federation

- 2. Other systems need access to data records or files stored in the PLM system**

## Web Services

# Federation - Definition



- ▶ **Core concept in Aras: Federation creates a wrapper around data objects in remote systems, and then extends their definition with PLM functionality**
- ▶ **The wrapper defines the data transfer, data mapping and the internal PLM representation**
- ▶ **We can add security, web interface, workflow routing to data in the legacy system, using it as a data source for a new audience of users**
- ▶ **We can federate an entire ItemType or just federate a few properties on a PLM internal Item in Aras**

# Federation – Definition (continued)



- ▶ **We aggregate data from many sources in this way in order to:**
  - Present the aggregated data in a single window for the user, e.g. a Mash-Up
  - Execute business rules, BI dashboards, or Reports against the consolidated data set
  - Add advanced PLM capabilities such as BPM workflow or a secure web interface to legacy systems

**Done well, the end-user is not aware of the sources of data**

# Federation Scenarios



## ▶ Corporate document management system

- Documents checked in to the PLM system are transferred directly to a corporate document archiving solution
- Searching and viewing documents is performed within the PLM interface, directly accessing remotely stored data

## ▶ Cost and inventory-on-hand fields in ERP

- Material cost may be maintained in the ERP system, but is useful information to engineers using PLM
- The PLM material forms contains a mash-up of data from both PLM and a real-time ERP query
- The PLM workflows may branch automatically based on the levels of inventory of parts being changed

# Why Server-Based Federation?



- ▶ Aras Innovator is an HTML Client; we can use the client as an integration platform also.
- ▶ Server-based Federation Advantages:
  - PLM client is sent a single data set, and is not aware that core business objects have been modified with remote properties, or that the data objects are not native
  - Integration of remote data objects is encapsulated on the server, eliminating client customizing
  - Server-side business rules such as Permissions, LifeCycle, and Workflow logic can act on the 3<sup>rd</sup> party data objects

# Simple Exercise



- ▶ Create an ItemType named Test with 1 Property=name
- ▶ Add a Server Event OnGet with a Method that creates and returns a hardcoded XML string

```
<Item type='Test' id='1234'><name>Hello</name></Item>
```

- ▶ Select this item on the TOC
- ▶ Grid will populate with the 'fake' data.

- ▶ **NOTE:**

- Had to create a dummy ID for each row

# Why is the Simple Exercise Simple?



- ▶ **Aras Client and all internal framework services expect AML to be returned on every transaction - this is key to the simplicity of Federation**
  
- ▶ **No customizations normally on the client**
  - As long as AML is returned, the client is happy to work with any data it receives
  - Exception is file check-in handling ( \*\* discussed later )
  
- ▶ **Use standard Server Events to add the Federation logic**
  - OnAfterGet                      to merge federated properties
  - OnGet                              to return 100% federated items

# Pseudo Code for a View



- ▶ **Assume “MyFederation” OnGet event - VB/C# Method**
- ▶ **Assume entire ItemType has been Federated**
- ▶ **Properties for the remote object, TOC Access, Form Views are defined in standard Aras Innovator ItemType Editor**
- ▶ **2 Modes:**
  1. Simple Searching and Viewing (unstructured)
  2. Using the Federated Item in Relationships  
For example: Part to Document Relationship where the Documents are federated

# Pseudo Code for View #1



- User query parameters from the Simple Search grid row are Properties on the Me / This object
- Check if this is a Get for one record or a set. **How?**
- Build the query for the remote system, passing the query parameters the user entered in the client
- Run the query code ( \*\* discussed next )
- Create an AML string from the query result
- Add a dummy ID for each row
- Return AML

**QUIZ: Is there an ItemType table for this item?**

# Pseudo-Code for View #2



**Federated Items will be used as Item Properties,  
or either the Source or Related in a Relationship**

**✓ ItemType is not flagged as Federated this time  
because we do want an Aras Innovator table**

**QUIZ - What properties must this table store?**

# Pseudo-Code for View #2



Three properties are required to maintain configurations

- ID
- Federated-ID
- Keyed-Name

**Solution:** in the OnGet method we add a function to register the data objects found via user queries, assigning a new Aras Innovator ID to each

*I use a small stored procedure to execute this efficiently*

# Remote Data Access Techniques



## ▶ Web Services

- Building an ASPX/PHP page as a wrapper around legacy systems
- Using Visual Studio to make a Proxy
- XMLHTTP object

## ▶ ODBC

## ▶ Commercial API from the other system vendor

**QUIZ: How do we register a DLL (either API or WS Proxy) so that you can use the DLL directly in Methods ?**

# Other Design Considerations



- ▶ **File check-in to a Federated source**
  - After check-in, move file from Aras vault to remote vault
  - Aras client does direct file upload
  
- **Actions for Add, Update and Delete are possible**
  - Synchronizing permissions and access

# Two Use Cases



1. Data in the ERP system or your legacy applications is relevant to the PLM users. We want this data and PLM data together in the right context on the PLM user interface -- like a Mash-Up

## Federation

2. Other systems need access to data records or files stored in the PLM system

## Web Services

# Web Services



- ▶ **Example: On-line website catalog needs data from the PLM system (drawings, specs, part numbers).**
  - BTW this is how the Aras web site runs
  
- ▶ **Shop floor MES system displays drawings from the PLM vault**

# Aras Training Class - Outline



- ▶ **Overview of Aras Web Services**
- ▶ **Creating a Web Service Configuration**
- ▶ **Generating the Web Service Code**
- ▶ **Publishing a new Application to IIS**
- ▶ **Testing the Web Service**
- ▶ **Consuming the Web Service example using Visual Studio**

# Web Services



- ▶ You can develop XML/SOAP messages that are sent directly to the Aras Server, the only requirement is that each transaction is authenticated, and you have the XML message payload correct.
  
- ▶ <http://myServer/Innovator/Server/InnovatorServer.aspx>
  - What? No WSDL. Why is there no WSDL?
  - WDL implies a static object model
  - Tradeoffs
    - Performance
    - Static vs. Dynamic view of the world

# Aras Web Services Choices



## ▶ Interfacing to the Aras Server directly

- InnovatorServer.aspx provides the ApplyItem service
- All AML commands are available
- No WSDL necessary

## ▶ Creating a Custom Web Service

- Advantages
  - Can generate WSDL
  - Can be used with tools that use WSDL to build interfaces
- Disadvantages
  - Static model
  - If an ItemType is altered, the service must be regenerated

# Service Publishing Overview



## ► Steps:

- 1) Establish a physical directory location for the service**
- 2) Create a Web Configuration Item**
- 3) Save the Web Configuration File to web service directory**
- 4) Generate the Web Service program code and configure service**
- 5) Add the Web Service to IIS**
- 6) Test the Service**

# Creating Web Service Configuration

- Viewers
- Web Service Configurations
- Workflow Maps



**Web Service Configuration**

**1** Name: Work Order

**2** Description: Work Order Service Definition

**3** Generate Wrappers For Value Types:

Item Types:  Work Order

**4**

**6** Properties

- cost
- instructions
- item\_number
- maintenance\_contract
- priority
- resolution
- response\_team
- scheduled\_date
- service\_rep [ User ]
- service\_rep\_email

**8** Show System:

Type Alias: **5**

Property Alias: **7**

# Specifying Actions



Properties Actions

- Add
- Get
- Edit
- Delete
- Update
- Lock
- Unlock

Add Custom Action:

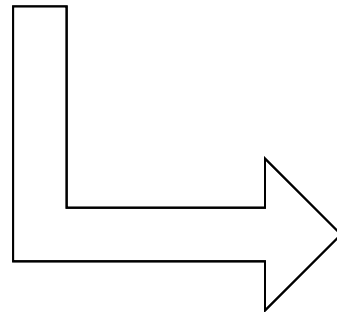
# Creating Custom Actions



## 1 Create Server Method

```
Method - WorkOrderCount - Windows Internet Explorer
Name WorkOrderCount Comment Test for web service
Server-side C#
01 Item itm = this.newItem("Work Order", "get");
02 itm = itm.apply();
03 int count = itm.getItemCount();
04 return this.getInnovator().newResult(count.ToString());
05
Checking syntax...
OK: 32CB0B2A7A9749583B2D646E0E37635EF.
Help Debug Ready ARAS INNOVATOR
```

## 2 Create Corresponding Action



Name	Type	Location
WorkOrderCount	Generic	Server

Label

Action

Method	Target	On Complete
WorkOrderCount	None	

Body

Item Query

Ready ARAS INNOVATOR

# Creating a Location Directory



- ▶ Locate the web service generator files:

*...Innovator/Server/bin/WebServiceGenerator*

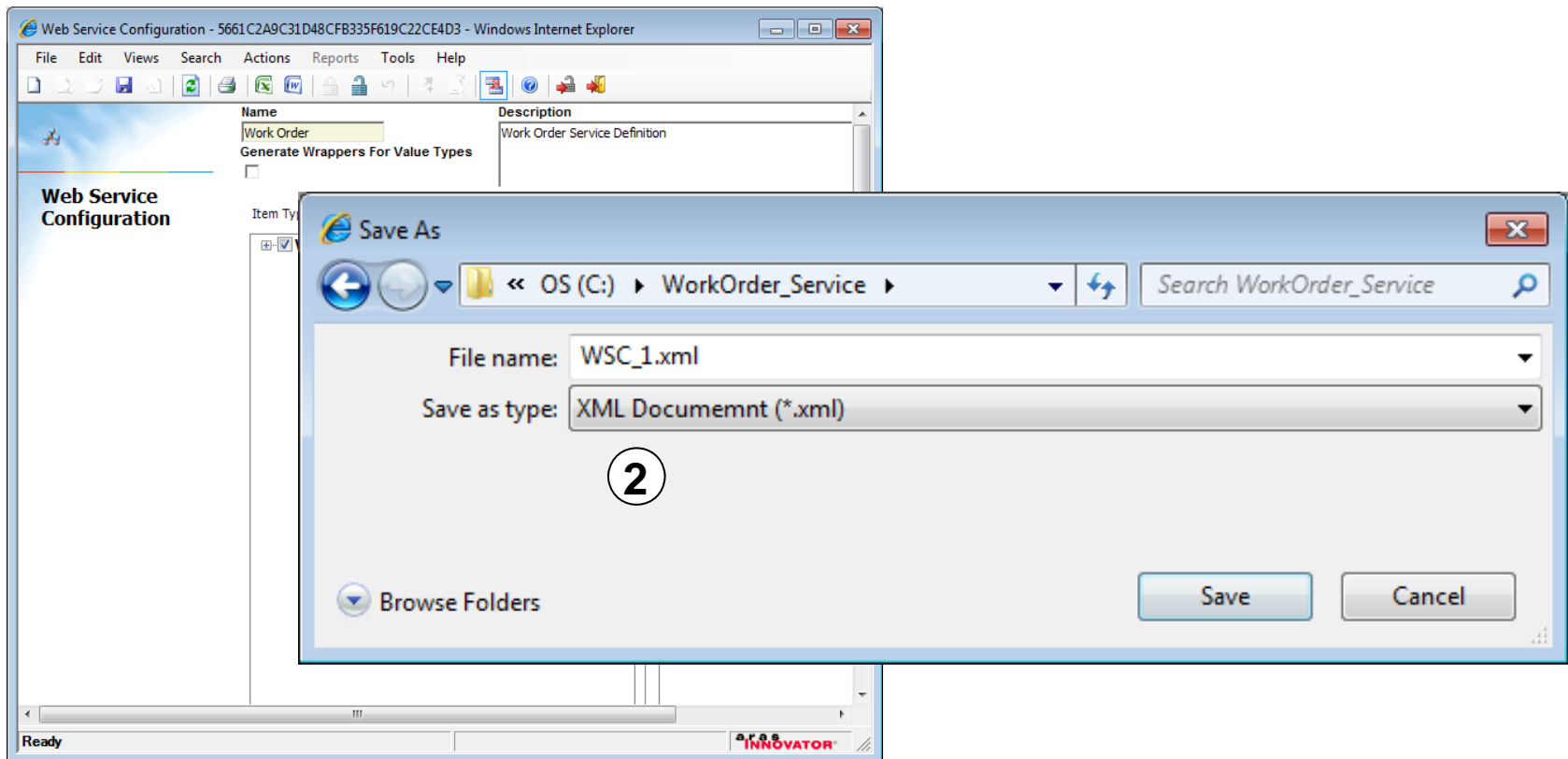
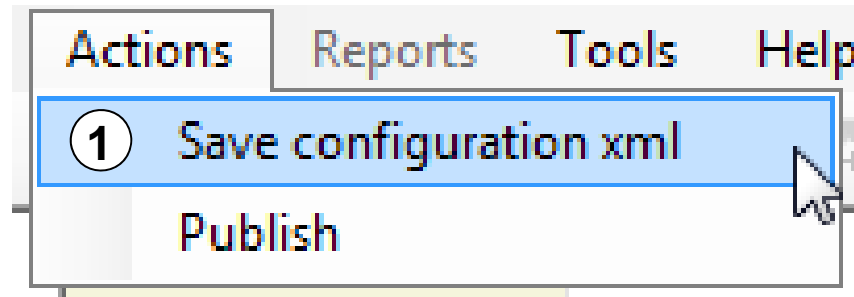
- ▶ Copy the folder to a new location

**Example:** C:\WebServiceGenerator

- ▶ Rename directory

**Example:** C:\WorkOrder\_Service

# Saving the Web Configuration File



# Generating the Web Service Code



## ▶ **GenerateWS.exe**

### ■ Parameters:

- Folder to generate Web Service code files
- File name of Web Service Configuration File
- Folder containing the template configuration .zip file
- Name of published web service

## ▶ **Example:**

```
GenerateWS . WSC_1.xml . WorkOrderService
```

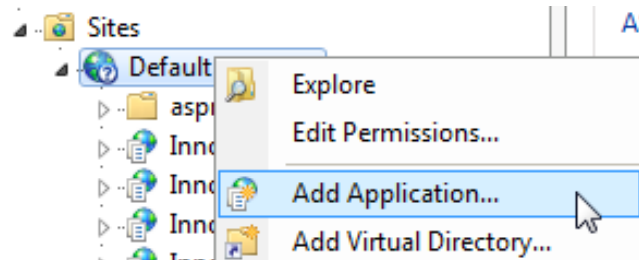
# Configuring the Web Service



- ▶ Locate the *innovator.config.xml* file
- ▶ Replace the following parameter values
  - URL
  - DB
- ▶ **Example:**

```
<?xml version="1.0" encoding="utf-8" ?>  
<appSettings>  
    <add key="url" value="http://localhost/Innovator920" />  
    <add key="db" value="DevelopingSolutions920" />  
</appSettings>
```

# Creating the IIS Application



**Add Application**

Site name: Default Web Site  
Path: /

Alias: WorkOrderService      Application pool: DefaultAppPool     

Example: sales

Physical path: C:\WorkOrder\_Service\InnovatorWebService     

Pass-through authentication

# Testing the Service



## ► Use Internet Explorer and enter the URL:

Example:

`http://localhost/WorkOrderService/innovator.asmx`

### WorkOrderService

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [ApplyAml](#)  
Applies AML.
- [ApplyMethod](#)  
Applies Method.
- [GetWork Order](#)  
Get items.
- [GetWork OrderByAml](#)  
Get items by aml criteria.
- [LogOn](#)  
Log on to Innovator.
- [WorkOrderCountWork Order](#)  
WorkOrderCount item.

# Consuming the Web Service in Visual Studio

A screenshot of the Visual Studio IDE. In the top-left corner, the 'References' tree is visible, with 'Web References' selected. A context menu is open over it, with 'Add Web Reference...' highlighted. The main window shows the 'Add Web Reference' dialog box. The URL field contains 'http://localhost/WorkOrderService/Innovator.asmx'. The 'Web services found at this URL:' section lists '1 Service Found: - Innovator'. The 'Web reference name:' field contains 'Innovator'. The 'Add Reference' button is highlighted in blue. The background shows the 'WorkOrderService' page with a list of supported operations: ApplyAml, ApplyMethod, GetWork\_Order, GetWork\_OrderByAml, LogOn, and WorkOrderCountWork\_Order.

References

- Web References
- Default.aspx
- Default.aspx.c
- Default.aspx.c
- Web.config

View in Browser

Browse With...

Convert to Web Application

Check Accessibility...

Add Web Reference...

reference to add all the available services.

Back

URL:  Go

## WorkOrderService

The following operations are supported. For a formal definition, please review the [Service Description](#).

- [ApplyAml](#)  
Applies AML.
- [ApplyMethod](#)  
Applies Method.
- [GetWork\\_Order](#)  
Get items.
- [GetWork\\_OrderByAml](#)  
Get items by aml criteria.
- [LogOn](#)  
Log on to Innovator.
- [WorkOrderCountWork\\_Order](#)  
WorkOrderCount item.

Web services found at this URL:

1 Service Found:

- Innovator

Web reference name:

Add Reference

Cancel

# Two Use Cases



- 1. Data in the ERP system or your legacy applications is relevant to the PLM users. We want this data and PLM data together in the right context on the PLM user interface -- like a Mash-Up**

## Federation

- 2. Other systems need access to data records or files stored in the PLM system**

## Web Services

# More Resources



## ▶ Federation test package

- Innovator-to-Innovator federation used for testing of the Server Events

## ▶ Aras wiki articles

- <http://www.aras.com/community/wikis/>

## ▶ Aras training classes and materials

- <http://www.aras.com/university/>



*ACE 2011 International*

# Understanding Federation and Web Services

---

[www.aras.com](http://www.aras.com)

