Web Service Nodes
Overview

- Summary
- What is a Web Service?
- What is a Web Service node?
- Use of a Web Service node
- Creating a Web Service node
- Executing a Web Service node
Web Service Node

◆ Web Service

◆ Call a method in a SOAP, SOAP WCF, or RESTful web service

◆ Must register the WSDL on the server

   ◆ Generates a client proxy dll in \Agility.Server.Web\Generated Assemblies\Webservice Proxies\

◆ Use a Web node to call a registered SOAP based web service
Web Service Node

- **User Credentials**
  - All automatic nodes run under the account running the TotalAgility Core Worker Service
  - If your web service has username and password credentials as parameters, these can be provided

- **EndPoint URL**
  - Test server – point at a test web service
  - Production server – point at a live web service (end point URL, won't find test URL so will look for end point URL)

- Use a RESTful activity if using RESTful web services
What is a Web Service?

- Web services are programmable units that can be accessed via the internet / intranet by disparate systems.

- Web services provide cross platform support and can take away the need to perform system integration (which speeds up development time).

- SOAP (Simple Object Access Protocol) based XML messages are normally used for communication purposes between clients and web services (and vice versa).

- SOAP is a standard XML format used to make remote procedure calls via the internet.
What is a Web Service?

- Web Services are normally run automatically by the TotalAgility engine using the identity of the core worker service (this can be changed).

- A URL to the .WSDL (Web Service Description Language) is required for any web service you try to call from the Platform.

- Try:
  - www.xmethods.com
  - www.webservicex.com

Note: Web services can only be used if the Microsoft .NET framework is installed on the TotalAgility server.
What is a Web Service node?

- A Web service node is an node that allows interaction with SOAP based internet web services.

- Web service activities depend heavily upon the broad acceptance of XML, HTTP and other Internet standards that support interoperability.
Why Use a Web Service?

**Advantages**
- There is widespread adoption of Web Services.
- SOAP allow applications written in different languages and deployed on different Platforms (e.g. Unix, MS, SUN) to communicate with each other over a network.
- Is being used to replace the traditional methods of remote communication such as DCOM / RMI and fits in better with SOA (Service Oriented Architecture).

**Disadvantages**
- While SOAP offers benefits in the world of interoperability, it comes at a price, i.e. performance degradation. A comparative study between SOAP and DCOM showed that (depending on the amount of data transferred), SOAP performance degradation can range from minor (2 to 3 times slower than DCOM) to major (twenty times slower than DCOM).
- Additional development effort is required for implementation of features missing from SOAP, such as security and state management. Approx 10% higher than DCOM.
Examples of Web Service

◆ Automatic validation of a credit card number via the internet

◆ Retrieval of delivery information from a transportation company for a dispatched order

◆ Creation of an TotalAgility job from your own application / product

◆ Call a web service that has been developed for your own application / product
Steps in Creating a Web Service Node

◆ Set up a web service reference

**Note:** the web service reference contains the details of the web service, such as the URL (Uniform Resource Locator), etc.

◆ Create a Web Service node

◆ Associating the required web service to the node
Web Service References

Support for:

- Traditional SOAP web services (.asmx extension)
- WCF Web services (.svc extension)
- RESTful web services
Adding a SOAP Web Service Reference

- On the Home page select Integration
- Select Web Service References
- Click Add
- Select SOAP
- Enter the web service Name and URL
- Optionally specify an EndPoint URL and/or User Credentials
Adding a Web Service Node

- Add an **Ordinary** activity
- Set the **Type** property to **Web Service**
- Select **Configuration**
- Select the **Integration** consume icon
- Select **Web Services**
- Drag and drop the web service reference into the consume area
- Select **Class** and **Method**
- Map variables as required
XML expression can be generated to extract the required data from the returned XML String.
Summary

- You should now be familiar with:
  - Web Service nodes and their use within TotalAgility
  - How to add a Web Service node