



▶ Microsoft .NET

▶ Writing Web Services

Introduction

A Web Service is a programmable class that may be called by any program running on a computer on the same LAN, or any program running on any computer on the internet.

Dyalog Web Services are hosted (i.e. executed) by ASP.NET running under Microsoft IIS.

A Web Service may expose a number of Methods and Properties. Methods may be called synchronously (the calling process waits for the result) or asynchronously (the calling process invokes the method, continues for a bit, and then subsequently checks for the result of the previous call).

Web Service (.asmx) Scripts

Web Services may be written in a variety of languages, including APLScript.

The first statement in the script file declares the language and the name of the service. For example, the following statement declares a Dyalog Web Service named GolfService.

```
<%@ WebService Language="apl" Class="GolfService" %>
```

Note that Language="apl" is specifically connected to the Dyalog APL script compiler through the ASP.NET system file Machine.config.

A Dyalog Web Service script starts with a :Class statement and ends with an :EndClass statement. These statements are directives used by the Dyalog script compiler and are specific to Dyalog.

The :Class statement declares the name of the Class (which must be the same as the name declared in the WebService statement) and the Base Class from which it inherits, which is normally System.Web.Services.WebService.

```
:Class GolfService: System.Web.Services.WebService
```

Following the :Class statement, there may appear any number of APL expressions and function bodies. Following these there must be a :EndClass statement. Internal sub-classes (nested classes) may also be defined within the main :Class ... :EndClass block.

Example Web Service

The following example eg1.asmx illustrates an APL Web Service. The file eg1.asmx is mapped to a URL (for example <http://localhost/apl.net/webservices/eg1.asmx>) via an IIS Virtual Directory.

Dyalog Ltd

South Barn
Minchens Court
Minchens Lane
Bramley
Hampshire
RG26 5BH
United Kingdom

Phone:

+ 44 (0) 1256 830 030

Fax:

+ 44 (0) 1256 830 031

e-mail:

sales@dyalog.com



- ▶ Microsoft .NET
- ▶ Writing Web Services

```
<%@ WebService Language="apl" Class="APLExample" %>

:Class APLExample:System.Web.Services.WebService

  ▽ R←Add args
  [1] :Access WebMethod
  [2] :ParameterList Int32 arg1, Int32 arg2
  [3] :Returns Int32
  [4] R←+/args
  ▽
  :EndClass
```

The *Add* function defined above is exported as a method that takes exactly (and only) two parameters of type Int32 and returns a result of type Int32.

Line [4] could in fact be coded as:

```
[4] R←args[1]+args[2]
```

because .NET guarantees that a client can only call the method by providing two 32-bit integers as parameters.

Testing APLExample from Internet Explorer

If you connect, using Internet Explorer (IE), to a URL that represents a Web Service, IE displays a page that displays information about the service and the methods that it contains. In certain cases, but by no means all, the page also contains form fields that let you invoke a method from the browser.

The screen shot next page shows the page displayed by IE5 when it is pointed at eg1.asmx. It shows that the Web Service is called *APLExample*, and that it exports a single method called *Add*. Furthermore, the *Add* method takes two parameters named *arg1* and *arg2*.

Dyalog Ltd

South Barn
Minchens Court
Minchens Lane
Bramley
Hampshire
RG26 5BH
United Kingdom

Phone:

+ 44 (0) 1256 830 030

Fax:

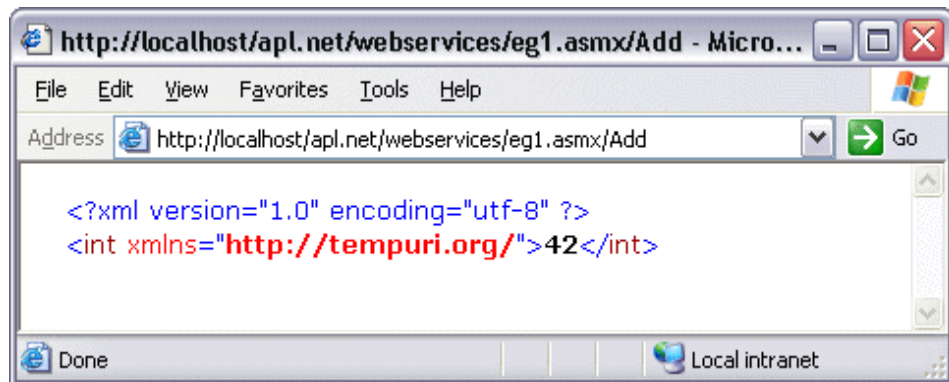
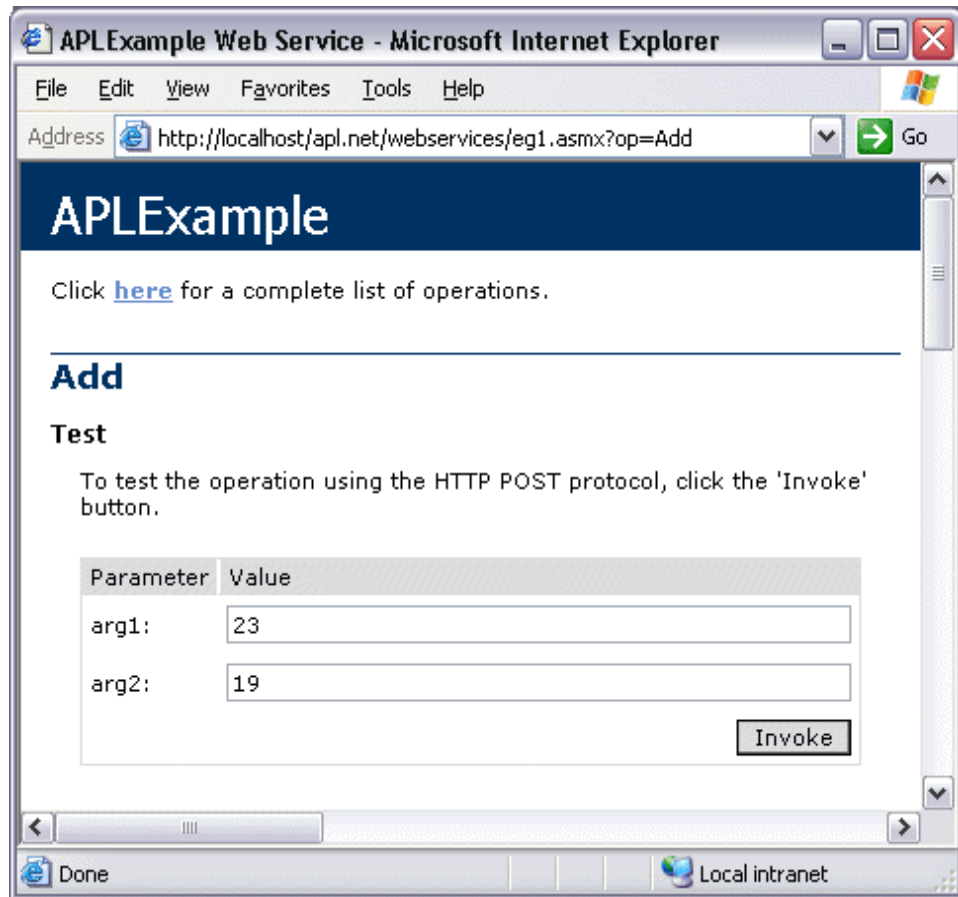
+ 44 (0) 1256 830 031

e-mail:

sales@dyalog.com



- ▶ Microsoft .NET
- ▶ Writing Web Services



Dyalog Ltd
 South Barn
 Minchens Court
 Minchens Lane
 Bramley
 Hampshire
 RG26 5BH
 United Kingdom

Phone:
 + 44 (0) 1256 830 030

Fax:
 + 44 (0) 1256 830 031

e-mail:
 sales@dyalog.com