WSDL2RPG – FAQ

FAQ What Do I Need to Know For Upgrading to The Latest Release

Status of this document

Date: 21.01.2015
Version: 1.1

Question

What do I need to know for upgrading to the latest release?

Answer

The answer to this question depends on the release you are on.

Version 1.14 or higher

If you are on WSDL2RPG v1.14 or higher, there is nothing special to consider when you want to upgrade to the latest release. Your generated stubs should run well with it.

At least all my old test programs run well with the latest release.

Version 1.12 or higher

If you are on WSDL2RPG v1.12 or higher, you need to know that with 1.14 WSDL2RPG had been changed to add the local time zone to all date, time and timestamp values.

Example:

```
before:      <start>2014-12-23</start>
after:       <start>2014-12-23+01:00</start>
```

That might be a problem in case you and the web service provider are in different time zones.

You can disable the local time zone by adding parameter **WSDL_NO_TIMEZONE** to the Marshallers in question:

```
'<tns1:date>' + Marshaller_toDate(
i_impl_echoDateTime.aDateTimeStructure.date : WSDL_NO_TIMEZONE ) +
'</tns1:date>' +
```

Version 1.11 or lower

For the older releases you need to modify your existing stubs by hand, if your stubs connect to RPC encoded Web Services, because with v1.12 I fixed some problems with **<MultiRef>** elements. These elements are used only by RPC encoded webs services. Hence you can safely upgrade to the latest release, as long as you do not call RPC encoded Web Services.

So the first thing you need to do is to check your Web Services. For that you need to open the wsdl file and search for the **<binding>** element:

```
<wsdl:binding name="whateverName" type="tns:whatEverNamespace">
```

Inside the `<binding>` element there should be a SOAP binding element that specifies the style of the Web Service. For RPG encoded Web Services the element should look like this:

```
<soap:binding style="rpc"
```

Web Services with a **'document'** encoding can be **safely ignored**.

For RPC encoded web service you need to add the following line to your generated stubs:

```
MultiRef_finalize();
```

Example:

```
              if (digest);
                 http_proxy_setAuth(HTTP_AUTH_MD5_DIGEST
                                      : userid: password);
              else;
                 http_proxy_setAuth(HTTP_AUTH_BASIC
                                      : userid: password);
              endif;
           endif;

           userid = '';
           password = '';
        endif;
     else;
        MultiRef_finalize(); // Required for RPC style services
        err = 0;
     endif;
  enddo;

  // Preserve parser error code and message text
  if (err = HTTP_RDWERR);
     g_errors.xmlError.code = XMLToolkit_getError(
                                 hSAXParser
                                 : g_errors.xmlError.text
                                 : g_errors.xmlError.line
                                 : g_errors.xmlError.column);
  endif;
```