BPEL

A Step by Step Guide: Model-Driven Generation with Enterprise Architect

This document will teach you how to use the Business Process Modeling Notation (BPMN 1.1) as a visual approach to producing BPEL code within Enterprise Architect. A "Hello World" example is used to illustrate how to generate BPEL and WSDL code automatically from BPMN.

Table of Contents

Introduction....3
What You Will Learn....3
What You Will Gain....3
Getting Started with Business Process Execution Language....4
BPEL Overview....4
BPEL in Enterprise Architect....4
Hello World BPEL....4
Creating a BPEL Process....5
Creating WSDL for the BPEL Process....7
Snapshot of the BPEL file....15
Snapshot of the WSDL file....16
Section Review....17

Introduction

Business Process Execution Language (BPEL) is used to define the execution of a business process using web services. Before you implement a business process using BPEL, a business analyst will typically define the process using an easy-to-read visual notation. An example of such a notation is the Business Process Modeling Notation (BPMN), which is an OMG standard and used in Sparx Systems Enterprise Architect to automatically generate BPEL.

What You Will Learn

After reading this document you will learn how to:

- create a BPEL Process and a corresponding WSDL.
- use BPMN 1.1 to represent business processes in a work flow.
- create a web service (Synchronous WSDL Messaging Schema).
- generate BPEL code, based on a BPMN 1.1 diagram in Enterprise Architect.

What You Will Gain

This report will allow you to:

- model and define business processes that interact with external entities.
- define business processes using an XML based language.
- develop a faster, model-based approach for generating BPEL from business process models.

This Quick Start Guide has been written under the following assumptions:

- You have a copy of Enterprise Architect installed
  (Ultimate or Business & Software Engineering).
- You have a basic understanding of BPEL, Web Services and Enterprise Architect.

For more information on BPMN go to:

http://www.bpmn.org/

Getting Started with Business Process Execution Language

BPEL Overview

Business Process Execution Language (BPEL) defines the behavior of a Business Process based on the interactions between the Process and external entities. This interaction occurs through Web Services and uses WSDL 1.1 to represent the interfaces exposed by the Process to the outside world.

BPEL in Enterprise Architect

Enterprise Architect uses BPMN 1.1 as a graphical front-end to capture BPEL 1.1 Process descriptions and uses the mapping specified in the BPMN 1.1 specification to generate BPEL.

Hello World BPEL

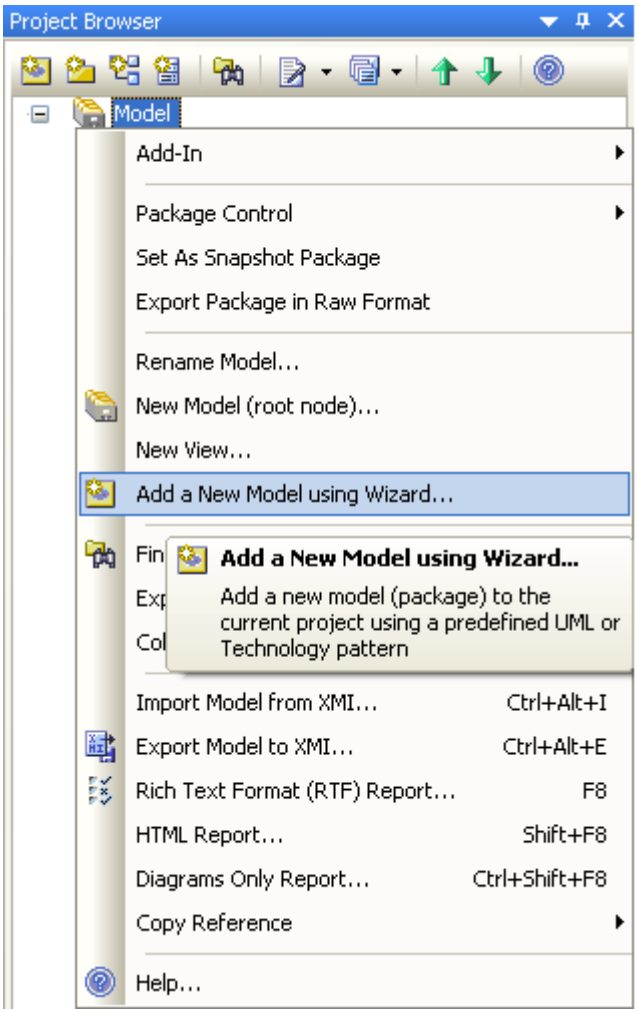
Let us create a simple Hello World BPEL, which will reply with the message “Hello World” when invoked. To create this BPEL, we need to:

1. Create a BPEL Process.
2. Create a WSDL for this BPEL Process.

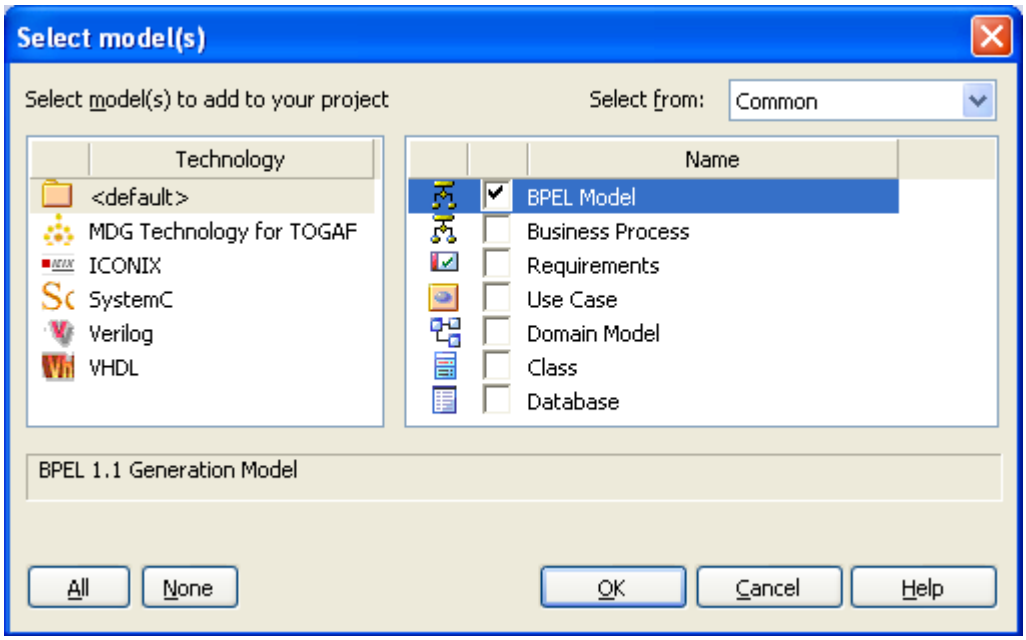
Creating a BPEL Process

We will use the built-in BPEL Model template as a staring point for creating our Hello World BPEL. This template will create a sample BPEL model structure, which can then be modified to suit our example.

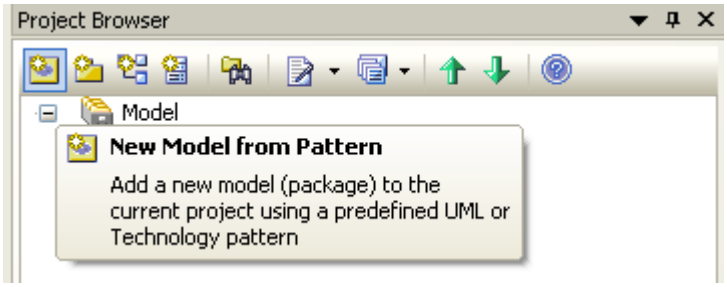
1. Open/Create an EAP file
2. Right-click on the root node in the Project Browser → Add a New Model using Wizard



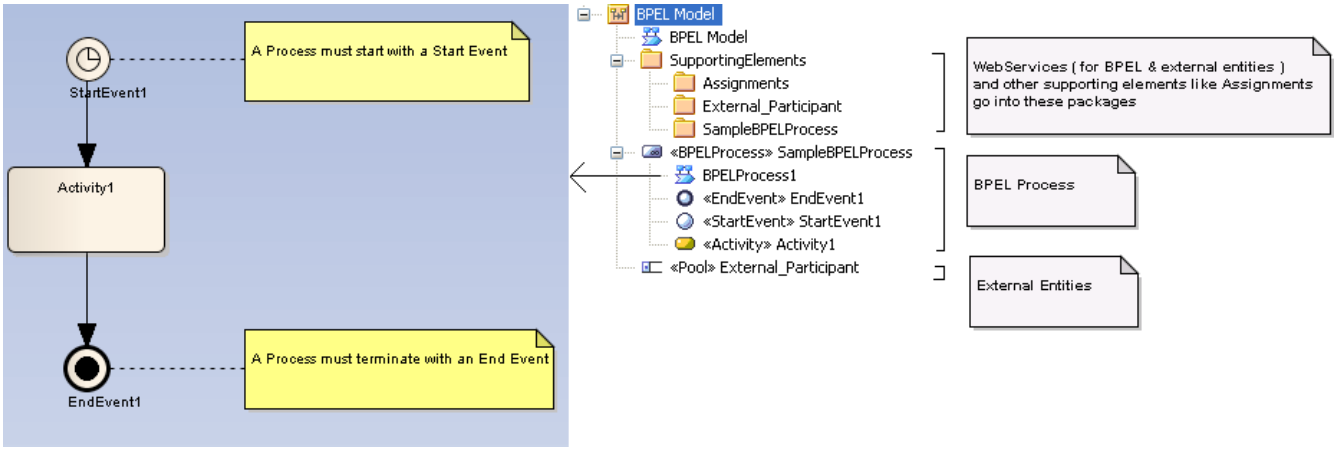
In the ***Select model(s)*** dialog, select ***BPEL Model***



NOTE *that the* ***Select model(s)*** *dialog can also be invoked by clicking on the* ***New Model from Pattern*** *button in the* ***Project Browser****.*



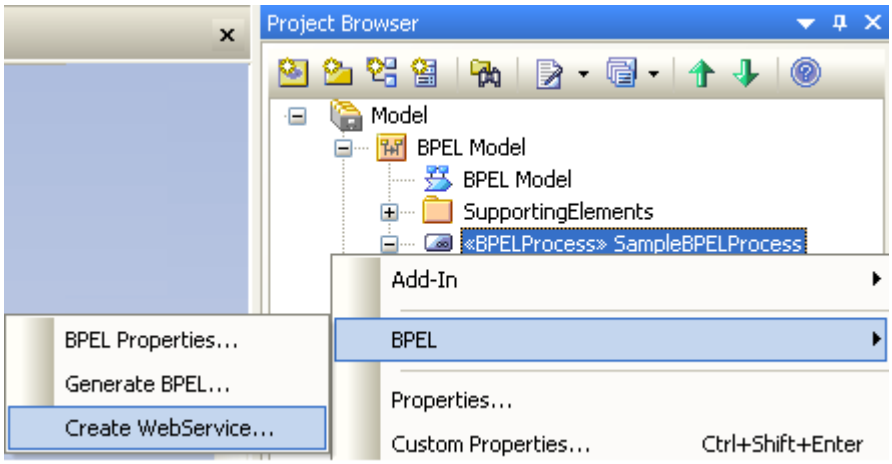
Enterprise Architect will respond by creating a package called ***BPEL Model***. This package will contain a BPEL package structure used in Enterprise Architect. The BPEL Process stereotyped element ***SampleBPELProcess*** represents the Process.



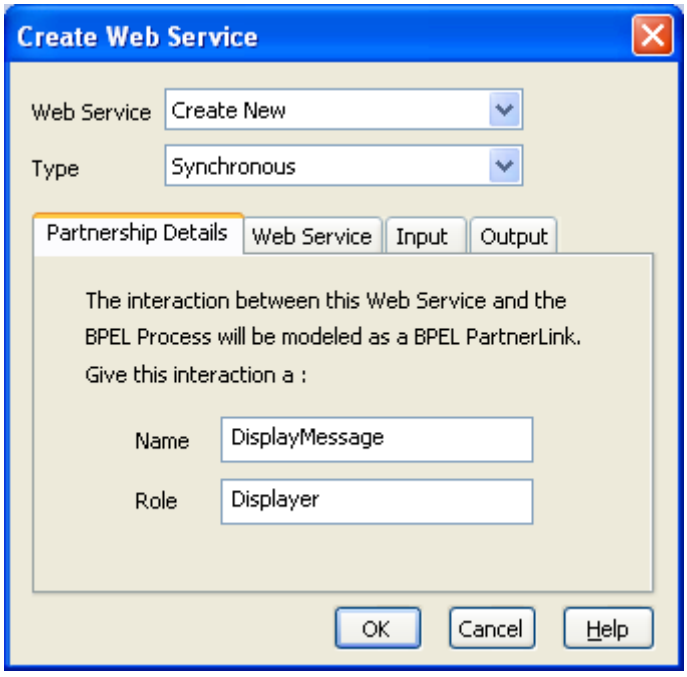
Creating WSDL for the BPEL Process

For the Hello World BPEL, let us model the BPEL such that a response is issued when an external entity invokes it. To model this scenario, we need to create a WSDL messaging schema that is Synchronous.

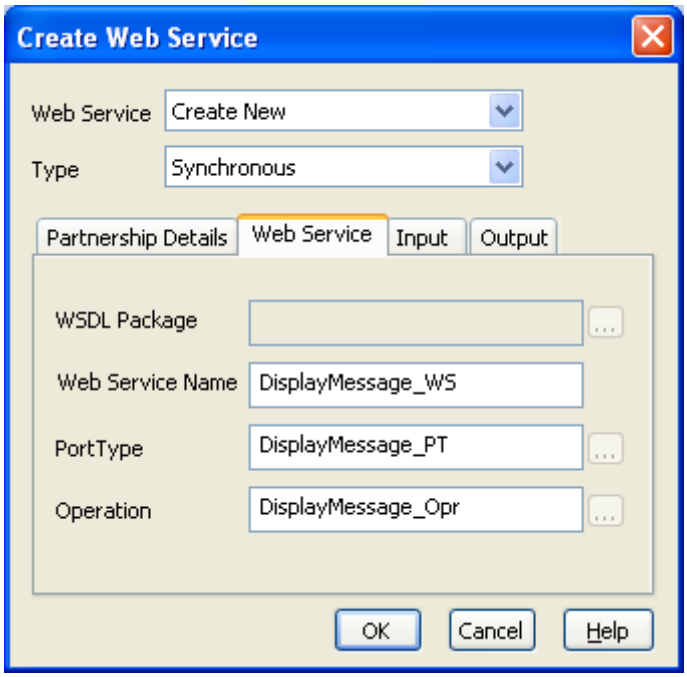
Right-click on ***SampleBPELProcess*** in the Project Browser → BPEL → Create WebService



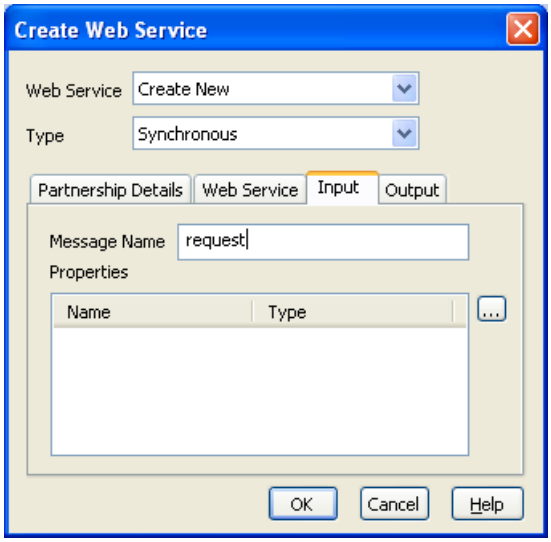
Set the values for the ***Partnership Details*** tab in the ***Create Web Service*** dialog as shown:



Set the values for the ***Web Service*** tab in the ***Create Web Service*** dialog as shown:

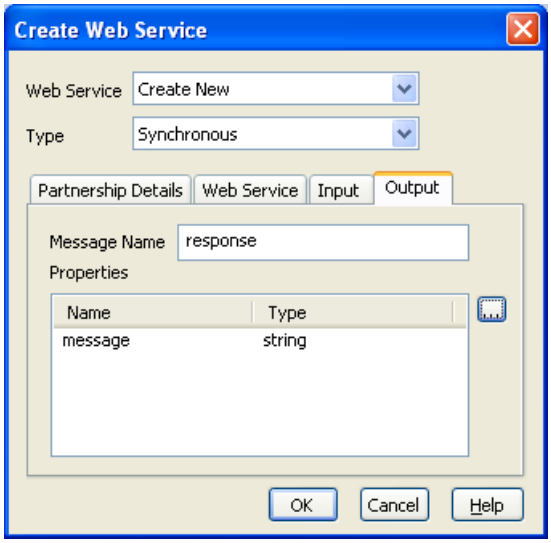


Set the values for the ***Input*** tab in the ***Create Web Service*** dialog as shown:



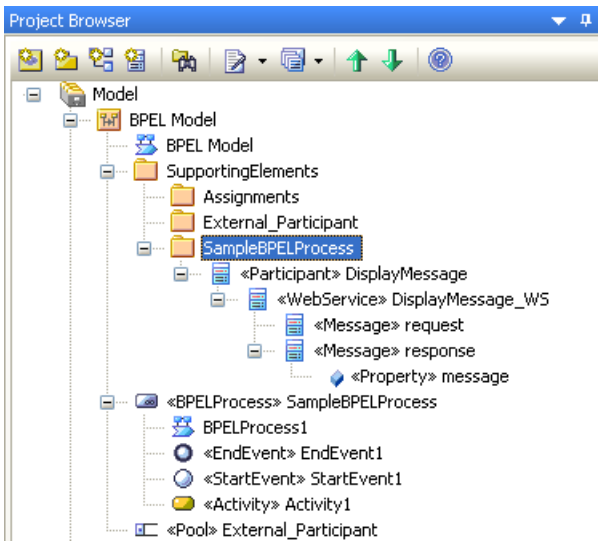
Note that since we are going to use this message to invoke the Process, we have not created any Properties for it.

Set the values for the ***Output*** tab in the ***Create Web Service*** dialog as shown:



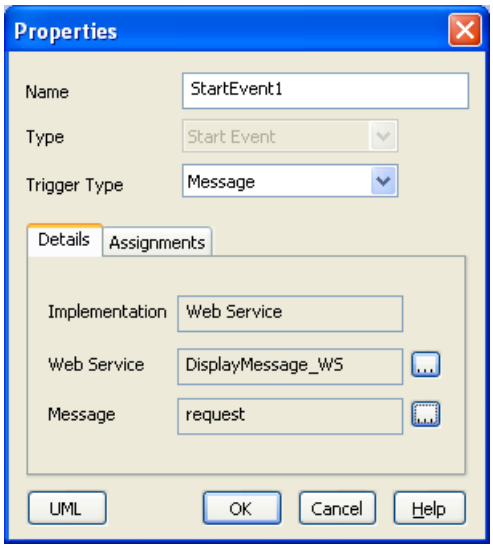
Since we are going to return a reply using this message, a Property called message is created which will hold a string - "***Hello World***" in our example.

Upon pressing press OK, Enterprise Architect will respond by creating a BPEL specific Web Service structure under the package ***SampleBPELProcess*** under ***SupportingElements***:

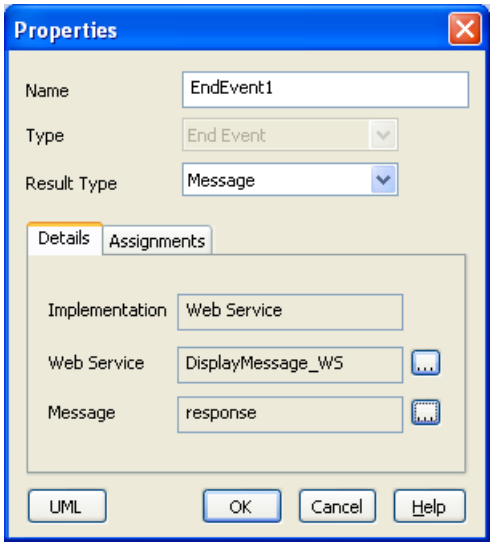


Modifying the BPEL Process to suit Hello World example

Now that we have defined our WSDL, let us modify the BPEL Process. Open the diagram ***BPELProcess1*** under ***SampleBPELProcess***. ***StartEvent1*** will start the BPEL Process when it receives a request message. To enable ***StartEvent1*** to receive a message, it should be of type ***Message***. So, double-click on this element and set its properties as shown:

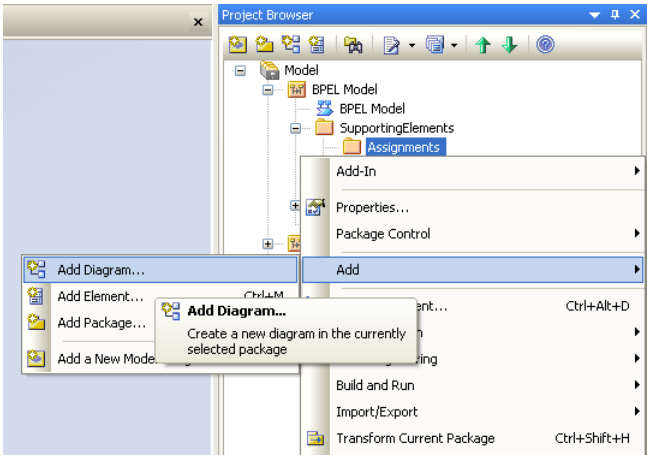


We will use ***EndEvent1*** to reply to the message. Just like ***StartEvent1***, ***EndEvent1*** should be of type ***Message*** in-order to respond with a message. So, double-click on this element and set its properties as shown:

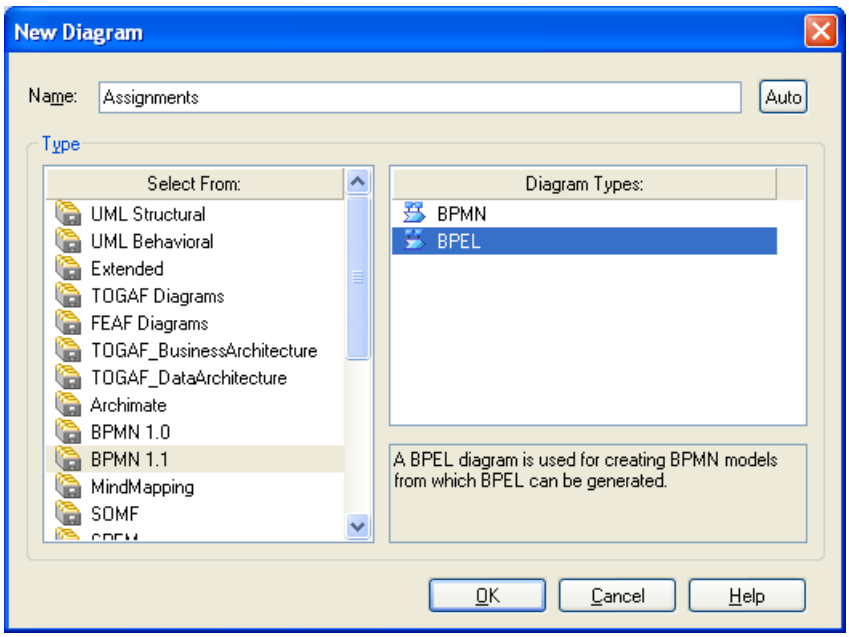


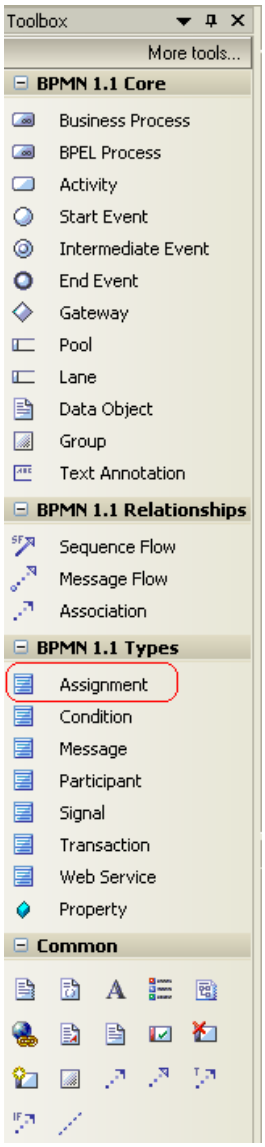
Let us use ***Activity1*** to assign the "**Hello World**" string to the response message. For this, we need to create an **Assignment** element and reference this element from ***Activity1***. Enterprise Architect follows a strict package structure and expects all the Assignments to be in the package ***Assignments*** under ***SupportingElements***.

Create a BPEL diagram under the package ***Assignments*** by right-clicking on the package → Add → Add Diagram:

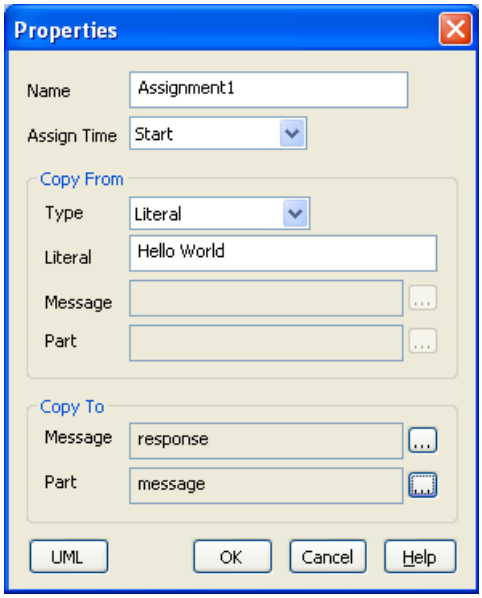


In the ***New Diagram*** dialog, select **BPMN 1.1** in the ***Select From:*** field and **BPEL** in the ***Diagram Types***: field. Enterprise Architect will respond by creating a ***BPEL*** diagram called ***Assignments*** under the package ***Assignments***. Drag-n-drop an ***Assignment*** icon from the **BPMN 1.1 Types** section of the BPMN tool-box into this diagram:

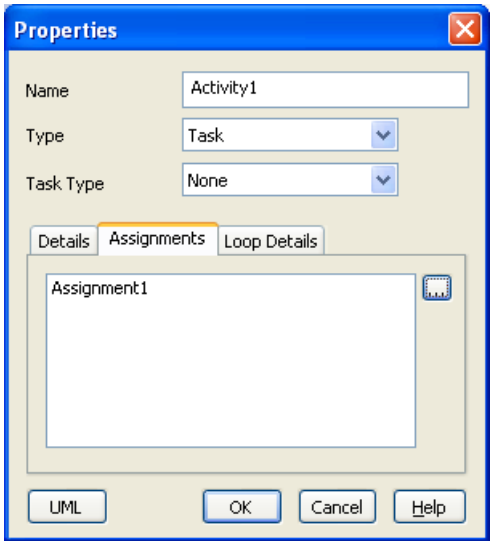




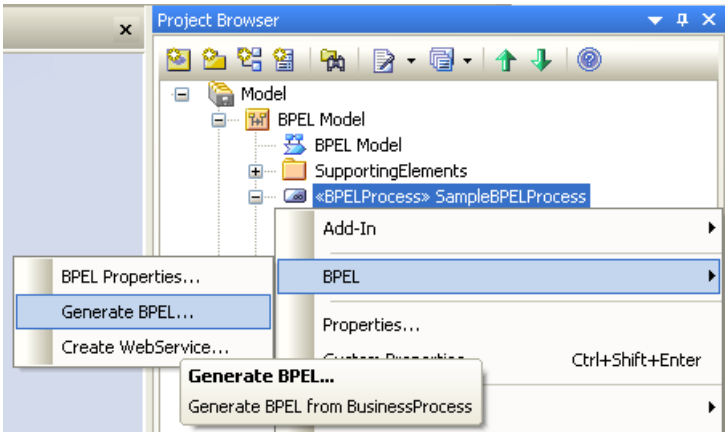
Set its values as shown:



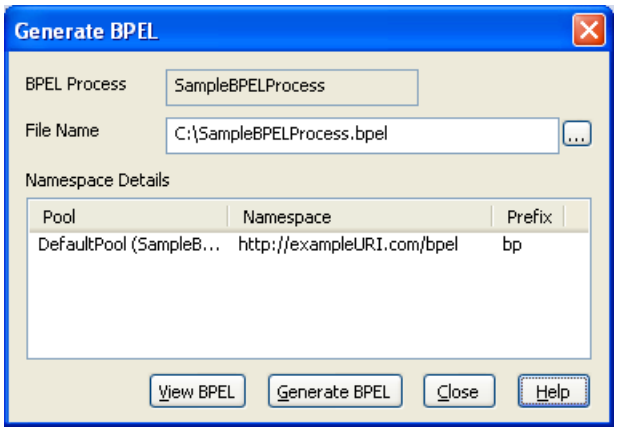
Back in the BPEL Process, double-click on ***Activity1*** and reference this Assignment as:



Now we are all set for generating BPEL. Open the ***Generate BPEL*** dialog by right-clicking on ***SampleBPELProcess*** → BPEL → Generate BPEL:



In the ***Generate BPEL*** dialog press the ***Generate BPEL*** button:



Enterprise Architect will respond by creating a BPEL and WSDL file.

Snapshot of the BPEL file

```
<?xml version="1.0"?>
<bpel:process name="SampleBPELProcess"
              abstractProcess="no"
              queryLanguage="XPath 1.0"
              suppressJoinFailure="no"
              enableInstanceCompensation="no"
              targetNamespace="www.samplebpelprocess.com"
              xmlns:bpel="http://schemas.xmlsoap.org/ws/2003/03/business-process/"
              xmlns:sb="www.samplebpelprocess.com"
              xmlns:bpw="www.samplebpelprocess.com/wsdl">

    <bpel:partnerLinks>
        <bpel:partnerLink name="MsgDisplayer" partnerLinkType="sb:MsgDisplayer_LT" myRole="MsgDisplayer"/>
    </bpel:partnerLinks>

    <bpel:variables>
        <bpel:variable name="request" messageType="sb:request"/>
        <bpel:variable name="response" messageType="sb:response"/>
    </bpel:variables>

    <bpel:sequence>
        <bpel:receive name="StartEvent1" createInstance="yes" variable="request"
                      partnerLink="MsgDisplayer" portType="sb:DisplayMessagePT" operation="DisplayMessageOpr"/>
        <bpel:sequence>
            <bpel:assign>
                <bpel:copy>
                    <bpel:from>Hello World</bpel:from>
                    <bpel:to variable="response" part="message"/>
                </bpel:copy>
            </bpel:assign>
            <bpel:empty name="Activity1"/>
        </bpel:sequence>
        <bpel:reply name="EndEvent1" variable="response" partnerLink="MsgDisplayer" portType="sb:DisplayMessagePT"
                    operation="DisplayMessageOpr"/>
    </bpel:sequence>
</bpel:process>
```

Snapshot of the WSDL file

```
<?xml version="1.0"?>
<wsdl:definitions name="SampleBPELProcess"
                  targetNamespace="www.samplebpelprocess.com"
                  xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
                  xmlns:sb="www.samplebpelprocess.com"
                  xmlns:plink="http://schemas.xmlsoap.org/ws/2003/05/partner-link/"
                  xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <wsdl:types/>

    <wsdl:message name="request"/>
    <wsdl:message name="response">
        <wsdl:part name="message" type="xs:string"/>
    </wsdl:message>

    <wsdl:portType name="DisplayMessagePT">
        <wsdl:operation name="DisplayMessageOpr">
            <wsdl:input name="request" message="sb:request"/>
            <wsdl:output name="response" message="sb:response"/>
        </wsdl:operation>
    </wsdl:portType>

    <plink:partnerLinkType name="MsgDisplayer_LT">
        <plink:role name="MsgDisplayer">
            <plink:portType name="sb:DisplayMessagePT"/>
        </plink:role>
    </plink:partnerLinkType>
</wsdl:definitions>
```

Section Review

What have we learned?

The BPMN notation is easily understood by business professionals, analysts and IT developers alike. You can use Enterprise Architect to leverage visual BPMN models to create executable web services that orchestrate and manage a series of business processes.

BPEL allows you to co-ordinate a number of different web services to achieve a particular business objective. The numerous benefits of using BPEL include platform and vendor independence, information exchange, common understanding amongst technical and business users, excellent orchestration and a flexible solution that can be easily scaled.

Enterprise Architect helps you improve productivity by automatically generating BPEL code, while also enhancing communication and providing greater understanding by using BPMN as an industry standard visual notation.