

# 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.



All material © Sparx Systems 2010 http://www.sparxsystems.com



### **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  $\rightarrow$  Add a New Model using Wizard

Project	: Brov	vser	<b>▼</b> ‡ ×				
9	2	3 🔒   🐜   🖻 • 🗐 •   🛧	10				
	<u>a r</u>	1odel					
		Add-In	+				
		Package Control	•				
		Set As Snapshot Package					
		Export Package in Raw Format					
		Rename Model					
	۵	New Model (root node)					
		New View					
	۵	Add a New Model using Wizard					
	Ca.	Fin 🔯 Add a New Model usin	o Wizard				
	6**6	Exr Add a new model (packad	e) to the				
		current project using a pr	edefined UML or				
		rechnology pattern					
		Import Model from XMI	Ctrl+Alt+I				
	鞋	Export Model to XMI	Ctrl+Alt+E				
	××∕ ××∕	Rich Text Format (RTF) Report	F8				
		HTML Report	Shift+F8				
		Diagrams Only Report	Ctrl+Shift+F8				
		Copy Reference					
	0	Help					



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

Select model(s)				
Select model(s) to add to your project	Select from: Common			
Technology	Name			
🦲 <default></default>	🕺 🗹 BPEL Model			
MDG Technology for TOGAF      ICONIX     SystemC     Verilog     WHDL	Business Process         Requirements         Use Case         Domain Model         Class         Database			
BPEL 1.1 Generation Model				
<u>A</u> ll <u>N</u> one	OK <u>C</u> ancel <u>H</u> elp			

**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  $\rightarrow$  BPEL  $\rightarrow$  Create WebService



С	reate Web	Servio	e		$\mathbf{X}$
	Web Service Type	Create Synchre	Create New		
	Partnership	Details	Web Service 1	Input Outpu	Jt
	The inte BPEL Pro Give this	raction t ocess wil ; interac	between this Wel I be modeled as a tion a :	b Service and a BPEL Partner	the Link.
	Na	me C	)isplayMessage		
	Ro	le C	Displayer		
			ОК	Cancel	Help

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:

Create Web	Servic	:e	X
Web Service Type	Create I Synchro	New 💌 onous 👻	
Partnership	Details	Web Service Input Output	_
WSDL Pack	age		
Web Service Name		DisplayMessage_WS	
PortType		DisplayMessage_PT	
Operation		DisplayMessage_Opr	
		OK Cancel <u>H</u> elp	



Create Web	Service	×
Web Service	Create New	
Туре	Synchronous 💌	
Partnership	Details Web Service Input Output	_
Message N Properties	Name request	
Name	Туре	
	OK Cancel H	elp

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:

Create We	b Service	×
Web Servic Type	e Create New 💌 Synchronous 👻	
Partnersh Message Properti	ip Details Web Service Input Output e Name response es	
messa	ge string	
	OK Cancel	Help

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:

Properties	×
Name	StartEvent1
Туре	Start Event
Trigger Type	Message 💌
Details Assignme	ents
Implementation	Web Service
Web Service	DisplayMessage_WS
Message	request
UML	OK Cancel <u>H</u> elp



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:

Properties	
Name	EndEvent1
Туре	End Event
Result Type	Message 🔽
Details Assignme	ents
Implementation	Web Service
Web Service	DisplayMessage_WS
Message	response
UML	OK Cancel <u>H</u> elp

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  $\rightarrow$  Add  $\rightarrow$  Add Diagram:

x	Project Browser	ř	🗢 ‡ X
	🙆 🖄 😫 🕯	😫  🗛 🛛 🖢 🖌 🕞 🖉 🖌 🗍 🔞	D
	😑 🍋 Mode	el	
	E	BPEL Model	
		😤 BPEL Model	
	<b>  -</b>	SupportingElements	
		Assignments	
		Add-In	•
	<b>E</b>	Properties	
		Package Control	•
😫 Add Diagram		Add	•
Add Element 😤 Add	сыдам Diagram	ent	Ctrl+Alt+D
Add Package Crea	te a new diagram	n in the currently	•
Add a New Mode	ted package	/ing	•
		Build and Run	•
		Import/Export	•
		Transform Current Package	Ctrl+Shift+H



Assignment

Condition

Message

Participant

Transaction Web Service

Property
 Common
 A

2

A 📰 🖻

🖬 🎦

고환 것의

Signal

E

Image: Control of the second secon

🎦 🔟

F.71



http://www.sparxsystems.com

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:

New Dia	ioram				Тос	lbox	🗢 🌣 🗙
New Die	Bram					N	fore tools
Mamor	A seizure ente			Auto	Ξ	BPMN 1.1 Cor	e
Na <u>m</u> e.	Assignments			Auto		Business Pro	cess
-Туре-						BPEL Proces:	5
	Select From:	^	Diagram Types:			Activity	
	UML Structural		😤 BPMN		0	Start Event	
	UML Behavioral		🕱 BPEL		0	Intermediate	Event
	Extended	=			0	End Event	
	TOGAF Diagrams					Gateway	
	FEAF Diagrams				~	Bool	
	TOGAF_BusinessArchitecture				u	P001	
	TOGAF_DataArchitecture					Lane	
	Archimate					Data Object	
	BPMN 1.0					Group	
	BPMN 1.1		A BPEL diagram is used for creating BPMN mo	dels	100	Text Annota	tion
	MindMapping		nom which BFEL can be generated.				stionshine
	SOMF	~				DPMN 1.1 Kei	acionsnips
					51	Sequence Fl	wc
						Message Flo	w
				lp	2	Association	
					Ξ	BPMN 1.1 Typ	es



Set its values as shown:

P	roperties	X
	Name	Assignment1
	Assign Time	Start 💌
	Copy From	
	Туре	Literal 💌
	Literal	Hello World
	Message	
	Part	
	Message	response
	Part	message
	UML	OK Cancel <u>H</u> elp

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

Properties		×
Name	Activity1	
Туре	Task 💌	
Task Type	None	
Details Assignments Loop Details		
UML	OK Cancel He	lp.



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



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

Generate BPEL		
BPEL Process	SampleBPELProcess	
File Name	C:\SampleBPELProcess.bpel	
Namespace Details		
Pool	Namespace Prefix	
DefaultPool (SampleB http://exampleURI.com/bpel bp		
View BPEL     Generate BPEL     Close     Help		

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



#### **Snapshot of the BPEL file**

```
<?xml version="1.0"?>
<bpel:process name="SampleBPELProcess"</pre>
             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"</pre>
                     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"</pre>
                    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.