SAP NetWeaver Process Integration 7.1
Providing Web Services in Java

SAP NetWeaver Regional Implementation Group
SAP NetWeaver Product Management
December 2007
Benefits

After reading this document, you will be able to understand:

- The Web services support in different SAP NetWeaver releases
- Composition Environment and Web services provisioning and consumption
- Enterprise Services Repository and Services Registry as the central place to design and publish services
1. **Introduction**
2. Composition Environment
3. Service Provisioning
4. Service Consumption
5. Summary
WEB SERVICES are  
SELF CONTAINED  
and SELF DESCRIBING  
APPLICATION FUNCTIONALITIES  
that can be  
PROCESSED  
through  
OPEN INTERNET STANDARDS  

There are two key elements in SAP’s definition of Web services:  

1) Application Functionalities  
   – SAP already owns a large number of interfaces that can be used as a basis for Web services.  
   – When starting the Web service project it was not possible to start with an entirely new model  
     for Web services and build a complete new application stack.  
   – Instead existing application functionality is taken and enabled as Web services – by simple  
     configuration, without any additional coding. There is a common concept for Java and ABAP,  
     i.e., it is possible to use BAPIs, EJBs, Java classes, IDocs, function modules etc. and turn  
     them into Web services.  

2) Open Standards  

Web services are based on XML.  

XML is the most widely accepted standard for data transfer between systems and over the internet.  
   It is  
   – vendor-independent,  
   – has no license fees associated with it  
   – and is thus available to everyone.  

The other important standards associated with Web Services are SOAP and WSDL, and they are  
both XML-based and therefore share these advantages.
Web Services Overview: The Nature of Web Services

Web Services

- act like a black-box that may require input and deliver a result
- work on top of any communication technology stack
- can be published, discovered and invoked based on open technology standards
- work in synchronous and asynchronous scenarios
- facilitate integration within an enterprise as well as cross enterprises
Web Service Description Language

Describes the basic characteristics of a Web service

- Supported operations and their data format  e.g. xCBL Order
- Supported protocols           e.g. SOAP
- Network address               e.g. http://a.com/order

Further information

- Specification:               http://www.w3.org/TR/wsdl12/

W3C definition:

“WSDL is an XML format for describing network services as a set of endpoints operating on messages containing either document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint. Related concrete endpoints are combined into abstract endpoints (services). WSDL is extensible to allow description of endpoints and their messages regardless of what message formats or network protocols are used to communicate”. (Note March 15, 2001)
Web Services Overview: The Paradigm

1. Create Web service
   - Create WSDL
   - Publish WSDL (opt.)

2. Discover Web service
   - Develop Client App

3. Web service Execution

Web Service Client

Services Registry

Direct WSDL exchange

Web service Provider

Business Functionality

WSDL Document

SOAP
Providing Web Services in SAP NetWeaver 2004

Inside-Out Approach

1. Implement Business Application
2. Generate Web Service
3. Activate / Deploy Web Service

Web Service Runtime

Development Tools

Service Definition

ABAP

Java class

Proxy

Business Application

UDDI

WSDL

SAP NetWeaver

WS Consumer

SOAP

Standard Interfaces

EAI

BAPI

RFC

SAP NetWeaver Process Integration 7.1
Providing Web Services in SAP NetWeaver 2007

Outside-In Approach

1. Define Service Interfaces
2. Generate Proxies
3. Implement code Backend
4. Register WSDL

Web Service Runtime
- Service Definition
  - Standard Interfaces
  - Business Application
- SOAP
- WSDL

Development Tools
- ABAP
- JAVA

WS Consumer

ES Repository

Services Registry

Java class
EBI
BAPI
Other Services
Consuming Web Services in SAP NetWeaver 2007

Web Service Provider

Business Functionality

Web Service

Soap

Web Service Runtime

Web Service Proxy

Web Service Consumer Application

Development Environments

ABAP

Java

Services Registry

ES Repository

WSDL
Working with Enterprise Services – ESR and Services Registry

1. Define Service
2. Generate Proxy
3. Implement
4. Publish

Non-SAP Implementation

Implementation in SAP Backend

Enterprise Services Repository

Services Registry

Providing Services

Consuming Services

1. Define Service
2. Generate Proxy
3. Implement
4. Publish

5. Discover
6. Compose
7. Deploy & Configure
8. Run

SAP NetWeaver Process Integration 7.1
Service Enablement: Step by Step

**Service Provisioning**

- **Define**
  - Metadata storage and Interface modeling in the Enterprise Services Repository
  - Creation and relationship settings of Service Interfaces, Operations and GDTs

- **Generate**
  - Proxy generation in ABAP and JAVA
  - Integrated development tools with ESR -> Enterprise Service Browser and WSDL import

- **Implement**
  - Business logic supported by EJB 3.0 (Java) and BAPIS (ABAP)
  - Web Service annotations supporting service provisioning from existing EJBs

- **Publish**
  - Services registry integrated with development tools
  - Service configuration and testing supported by WS Administrator and WS Navigator

- To go with Eclipse proved to be the best possible choice – practically, Eclipse eliminated the Java tools market
- SAP deployment comes smoothly integrated into the Eclipse tool-set
Service Enablement: Step by Step

Service Consumption

**Discover**
- Development tools in ABAP and JAVA integrated with Services Registry
- Web Dynpro and Visual Composer wizards part of CE 7.1

**Compose**
- UI composition supported by IDE perspectives
- Process composition supported by Guided Procedures

**Deploy & Configure**
- Configuration and deployment done by IDE, NetWeaver Administrator and Engine tools

**Run**
- Client applications running on Interactive Adobe Forms, Flex and Web Dynpro.
- “Point to point” and “Brokered” runtime scenarios supported.

- To go with Eclipse proved to be the best possible choice – practically, Eclipse eliminated the Java tools market
- SAP deployment comes smoothly integrated into the Eclipse tool-set
1. Introduction
2. Composition Environment
3. Service Provisioning
4. Service Consumption
5. Summary
A Business Process Platform - What’s in it?

Major Building blocks of a Business Process Platform

- **... a composition environment** to plan and build flexible and innovative business processes
- **... an extensible repository of enterprise services definitions**
- **... process components** that implement those services in a robust and reusable fashion
- **... an integration platform** to integrate and run service-enabled business applications
- **... and an entire Ecosystem that delivers innovation with it**

With a business process platform, organizations can respond rapidly to change and drive business process innovation while reducing cost at the same time.

It allows companies to efficiently plan, build, run - standard and new business processes (delivered by the ecosystem) while providing a coherent application and integration infrastructure that enables:

- broad user adoption
- adaptable processes
- trusted information
- …while optimizing IT

A Business Process Platform is the prerequisite for the controlled and cost-effective deployment of enterprise SOA.

A Business Process Platform consists of

- an extensible repository of enterprise services definitions,
- application logic that implements those services in a robust and reusable fashion,
- a composition environment to plan and build flexible and innovative business processes
- an Integration platform to integrate and run service-enabled business applications
Business Process Platform: Composition Environment

Service Enablement ➔ From Provisioning to Consumption

Enterprise SOA Infrastructure for:
- Service Provisioning
- Service Consumption
- Service design governance
Model-Driven Development and Deployment With SAP NetWeaver CE 7.1

- Robust, enterprise-class Java EE 5 application server
- Guided procedures (GP) for collaborative processes
- SAP Composite Application Framework (CAF) business object modeling and service composition
- Web Services / Enterprise Services Connectivity
- Enterprise Services Repository and Registry (ESR)
- Software lifecycle management and logistics (NWDI)
- SAP NetWeaver Developer Studio: Eclipse based development and modeling environment
- SAP NetWeaver Visual Composer for model-driven UI development (incl. analytics and voice)
- Web Dynpro Java
- SAP Interactive Forms by Adobe
- Federated Portal Network

Processes
- Event Flow Role

Views
- Portal and Web Analytics Forms Mobile and Voice

Services
- New service Service façade Basic service Data service Enterprise service

Bus
- Data or File
Provides the scalable, reliable, and productive enterprise SOA infrastructure to build, modify, and run innovative and flexible composite applications

Builds on the industry standard development environment from Eclipse and integrates composition tools

Enables standard enterprise services development and provisioning using Java EE 5 (EJB 3.0) and web services standards

Reduces time-to-value and development efforts through model-driven composition using an integrated set of modeling tools for services, views, and processes

Facilitates out-of-the-box enterprise services and SOA asset reuse from a stable business process platform

Provides a comprehensive software lifecycle management across entire composition stack to reduce TCO
1. Introduction
2. Composition Environment
3. Service Provisioning
4. Service Consumption
5. Summary
Service Provisioning

Define Generate Implement Publish

Service Definitions
- Integration Repository of PI 7.1 as the foundation of a central Enterprise Services Repository
- Based on SAP’s process components
- Integrated modeling environment
- Creation and relationship settings of Service Interfaces, Service Operations, Interface Patterns and GDTs
- Enhancements to well known objects and introduction of new ones
- Mass import of external definitions

- To go with Eclipse proved to be the best possible choice – practically, Eclipse eliminated the Java tools market
- SAP deployment comes smoothly integrated into the Eclipse tool-set
  - new, much easier client user interface
Service Provisioning

Define | Generate | Implement | Publish

Transition
- Integration Repository -> ES Repository
- Message Interface -> Service Interface

Enhancements
- Service interfaces may have several operations
- Interface Patterns and Operation Patterns
- Matching service interfaces
- Integrated modeling environment
- New data types
Generate Java Proxies
- Connect to ESR and browse for service definition
- Download WSDL and generate Java skeleton code
Implement business logic

- Complete your Java code in the skeleton code
- Deploy EAR to server and test web service
Service Provisioning

Define ▶ Generate ▶ Implement ▶ Publish

Optionally configure services at design time

- The configuration you apply via annotations in the SAP NetWeaver Developer Studio serves as the basis for the runtime configuration settings you can apply in the SAP NetWeaver Administrator.

- For example: `@SrPublication (location="<path>" )` – adds classifications to Web services

Configure runtime configuration

- Complete runtime configuration in SAP NetWeaver Administrator
- Configure both individual Web services and groups of Web services
Service Provisioning

Define ➜ Generate ➜ Implement ➜ Publish

WSDL++

Easier provisioning of Web services
- Java EE 5 compliant – easier to code!
- EJB 3.0, JAX WS 2.0, standard annotations and SAP annotations with reasonable defaults (including code completion)
- Code generation from ESR, WSDL, and SR
- Use Composite Application Framework (CAF) to integrate and enhance existing services and business objects, as well as to create new services
- Support for bottom-up approach for service-enabling existing applications
1. Introduction
2. Composition Environment
3. Service Provisioning
4. **Service Consumption**
5. Summary
Service Consumption

Discover via browse & Search
- ESR Browser integrated into CE Development Tools
- Services Registry Query integrated into CE Development Tools
- ESR and Services Registry accessible from Web

Discover
Compose
Deploy & Configure

Enterprise Services Repository

Services Registry

WSDL

Services Consumption

SAP Dev Tools
**Service Consumption**

- **Discover**
- **Compose**
- **Deploy & Configure**

### Resolve Collisions
- Automatic Collision Resolution
- Supply JAX-WS customization file
- Interactively resolve collisions

### Generate
- Generate Service Endpoint Interface
- Generate Java Types

### Consume
- Instantiate the generated stub
- Call Web Service methods on the generated stub
## Service Consumption

**Discover**

**Compose**

**Deploy & Configure**

### Consume
- Fully Automated Point and Click within CAF, Web Dynpros, Visual Composer & Guided Procedures – Generates Basic Model
- WS Need not be provisioned
- Adaptive – Connectivity is handled at runtime

### Create Model
- CAF: Enhance Model – similar to session bean concept in J2EE
- Web Dynpros: Create Model Directly within Web Dynpros
- Visual Composer: Automated

### Define Actions, Events and Rules
- Define logic within controllers and views in Web Dynpros
- Visual Composer: Rudimentary Support
- Guided Procedures

### Create User Interaction
- Create Views and Controllers in Web Dynpros
- Visual Composition in VC
- GP uses WD, VC or Adobe Forms
## Service Consumption

### Discover
- CAF, Web Dynpro, Visual Composer & Guided Procedures: Generate Web Service Model - Fully Automated Point and Click
- JEE 5: Import WSDL into workspace
- Service Definition might be abstract – connectivity is handled at runtime

### Compose
- CAF: Create Application Services & Business Objects
- Web Dynpro: Create Model Directly within Web Dynpros
- Visual Composer: Automated
- JEE 5: Generate JAX-WS Stubs

### Deploy & Configure
- Web Dynpro: Define controllers in MVC pattern
- Visual Composer: Rudimentary Support
- Guided Procedures – Define process workflow
- JEE5: Use JEE Toolset to put logics inside EJBs, Servlets...

### Logging & Tracing
- CAF, Web Dynpro, Visual Composer & Guided Procedures: Generate Web Service Model - Fully Automated Point and Click
- JEE 5: Import WSDL into workspace
- Service Definition might be abstract – connectivity is handled at runtime

### Automatic Authentication
- CAF: Create Application Services & Business Objects
- Web Dynpro: Create Model Directly within Web Dynpros
- Visual Composer: Automated
- JEE 5: Generate JAX-WS Stubs

### Define Actions, Events and Rules
- Web Dynpro: Define controllers in MVC pattern
- Visual Composer: Rudimentary Support
- Guided Procedures – Define process workflow
- JEE5: Use JEE Toolset to put logics inside EJBs, Servlets...

### Create User Interaction
- Web Dynpro: Create Views
- VC: Visual Composition
- GP: uses WD, VC or Adobe Forms
- JEE5: JSF, JSP, Servlets
Service Consumption

Discover  Compose  Deploy & Configure

Deploy and then Configure...
- Destinations
- Automatic Authentication & Security settings
- Logging & Tracing
- HTTP Transport Settings
- Web Service Addressing
- State Management
Agenda

1. Introduction
2. Composition Environment
3. Service Provisioning
4. Service Consumption
5. Summary
Summary – Providing Web Services in Java

- End to end integrated SOA based development in Java
- Central ESR for designing service definitions
- Central UDDI v3.0 registry for service consuming
SAP is Open

OPEN SOURCE
Leverage open source within SAP environments

BUSINESS STANDARDS
Driving standards for flexible business processes

TECHNOLOGY STANDARDS
Enabling SOA and platform interoperability