

Notas técnicas - Tips de SAP / ABAP – Tip en detalle Nro. 17

(Lo nuevo, lo escondido, o simplemente lo de siempre pero bien explicado)

Los nuevos escenarios de programación con SAP Netweaver (serie de varios tips)

”Web Services en el entorno ABAP: un caso práctico Paso a Paso”

Tema: ABAP Web Services, SAP Netweaver Web Application Server.

Descripción: El objetivo de este tip es introducir al lector en la **tecnología de web services** desarrollados desde el entorno nativo de SAP a través de un ejemplo práctico, así como evidenciar las ventajas de la utilización de web services en las instalaciones actuales como puerta de entrada a **Enterprise Service Architecture**.

Nivel: Avanzado

Versión: SAP WAS 6.40

Fecha pub: Diciembre de 2005

NOTA: Para una correcta comprensión de este artículo, recomendamos la lectura del tip anterior “**ESA y Web Services en SAP Netweaver: Introducción**”. Consulte nuestro sitio <http://www.teknodatips.com.ar/sap-netweaver.html>

*”Tips en breve/Tips en detalle” se envía con frecuencia variable y absolutamente **sin cargo** como un servicio a nuestros clientes SAP. Contiene notas/recursos/artículos técnicos desarrollados en forma totalmente objetiva e independiente. Teknoda es una organización de servicios de tecnología informática y **NO comercializa hardware, software ni otros productos**. Si desea suscribir otra dirección de e-mail para que comience a recibir los tips envíe un mensaje desde esa dirección a sapping@teknoda.com, indicando su nombre, empresa a la que pertenece, cargo y país.*

Tabla de contenido

- I. Introducción/Resumen Ejecutivo.
- II. ABAP Web Services: un caso práctico
 - Prerrequisitos
 - Creación de un Web Service
 - Creación de la Interfaz Virtual
 - Creación de la Definición del Web Service
 - Configuración del Web Service
- III. Consideraciones sobre Seguridad de Web Services
- IV. Dónde Obtener información adicional

I. Introducción/Resumen Ejecutivo

Web Services: El Contexto

Como desarrolláramos en el tip anterior (“ESA y Web Services en SAP Netweaver”), Web Services es hoy uno de los conceptos que más atención concentra en la industria del software. Los escenarios de negocio cada vez más competitivos y complejos, han gestado la demanda de funcionalidad cada vez más elaborada, y una oferta de soluciones cada vez más especializada, alimentada por el suceso de las aplicaciones tipo ERP. Sin embargo, la realidad muestra que no es posible implementar toda esta funcionalidad utilizando una única tecnología. Una infraestructura moderna de software debería permitir integrar funciones desarrolladas interna o externamente, por múltiples proveedores y plataformas en un único proceso completo y efectivo.

Hasta hoy, la integración de software se apoyaba principalmente en la **definición manual de interfaces**, formato de mensajes, y la **concordancia explícita** entre los “socios” de negocio involucrados. Este enfoque, además de laborioso, genera enormes trabas en la flexibilidad de las interfaces obtenidas y su capacidad de adaptarse a los cambios.

El término Web Service define a la vez una entidad de software y un modelo de programación, que, basado en estándares abiertos y universalmente aceptados, permite combinar funcionalidad implementada en las más variadas plataformas. En blanco y negro, un Web Service es una entidad ejecutable de software, totalmente encapsulada y “autosuficiente”, que puede ser detectada e invocada a través de una red para cumplir con un propósito determinado. El “consumidor” de un Web Service desconoce la complejidad interior de la misma, desconoce la plataforma y el lenguaje en el que está escrita, pero sí conoce el efecto concreto de su ejecución. La invocación del Web Service se realiza a través de protocolos de mensaje absolutamente estandarizados, y, por supuesto, los Web Services son reusables.

Los Web Services proveen un mecanismo para que las aplicaciones hablen unas con otras prescindiendo de la plataforma de cada una de ellas, y sin estar controladas por ningún proveedor. Una compañía podría, por ejemplo, diseñar un Web Service que permita verificar el historial crediticio de un cliente. Este Web Service podría estar disponible para ser “consumido” por cualquier aplicación interna o externa, sin importar si la misma corre en Linux, Windows, JAVA o ABAP.

SAP y Web Services

SAP brinda la posibilidad de extender las soluciones empresariales por medio de web services JAVA o ABAP. En caso de implementar procesos que involucran diversos socios comerciales y sistemas heterogéneos, puede ser beneficioso optar por desarrollos en JAVA. En cambio, para soluciones centradas en la selección de información, podemos considerar el entorno nativo.

El marco de desarrollo consiste de:

- Entorno de desarrollo de la personalidad ABAP
- Entorno de desarrollo de la personalidad J2EE
- Herramientas que soportan registración UDDI
- Entorno de ejecución distribuido SOAP (ABAP / J2EE)

Cualquiera sea la opción escogida contaremos con el siguiente soporte para la creación y administración de Web Services:

- Interfaces Virtuales: Es posible definir múltiples vistas de una implementación y publicarlas separadamente, así como renombrar u ocultar operaciones y parámetros, o definir valores estándar para los mismos.
- Definición y Configuración de WS: Es posible definir el tipo de comunicación y nivel de autenticación adaptando en función de las características técnicas del application server.
- Seguridad: SAP WAS soporta encriptación de datos a través de HTTP usando Secure Socket Layer Protocol (SSL).

- Publicación de Web Services en UDDI: Podemos publicar los Web Services como Business Services y ponerlos a disposición a través de la red.
- Web Service Homepage: En caso de contar con un motor J2EE instalado en el Application Server podemos acceder a la homepage del Web Service.

En este tip expondremos un ejemplo de **creación de un Web Service en ABAP**, evidenciando las ventajas de dicha opción, es decir, la posibilidad de basarnos en funciones RFC-enabled **ya desarrolladas y testeadas**, o en BAPIs de SAP, la familiaridad con el entorno de desarrollo, la facilidad que nos proveen las nuevas transacciones y el Wizard para la administración de Web Services, entre otras.

Para una mejor comprensión de este tip, sugerimos además la lectura del tip *Nro. 10: Entendiendo el SAP Web Application server desde el punto de vista del programador*.

II. ABAP Web Services: un caso práctico

Prerrequisitos

Para desarrollar exitosamente este ejemplo se debería tener instalado **SAP WAS 6.40**. Si bien el soporte de web services en SAP existe desde SAP WAS 6.20, **sólo la versión 6.40 implementa las transacciones y el wizard descritos en el ejemplo.**

Otro requisito es contar con J2EE Engine para poder acceder a la homepage del Web Service creado. En caso de no contar con dicho motor, podremos igualmente desarrollar el Web Service, pero deberíamos “consumirlo” desde una aplicación específica desarrollada en cualquier plataforma que soporte web services.

Creación de un “ABAP Web Service”

Es posible crear Web Services partiendo de las siguientes funcionalidades:

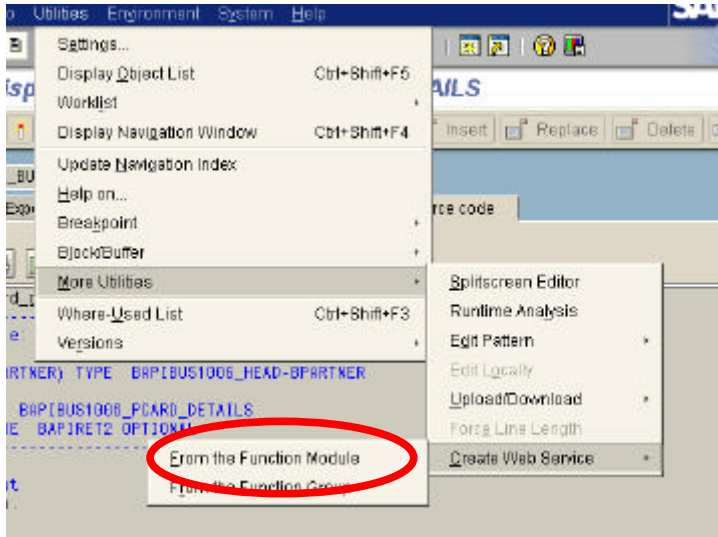
- Módulos de Función RFC-enabled
- Grupos de Función que contengan al menos una función RFC-enabled
- BAPIs
- Interfaces de Mensajes XI (Exchange Infrastructure)

Para nuestro ejemplo construiremos Web Services basados en BAPIs provistas para administrar la información de los Business Partner (BP), es decir, las personas físicas o sociedades con las cuales la empresa tiene algún tipo de relación de negocios. Las BAPIs que utilizaremos son las siguientes:

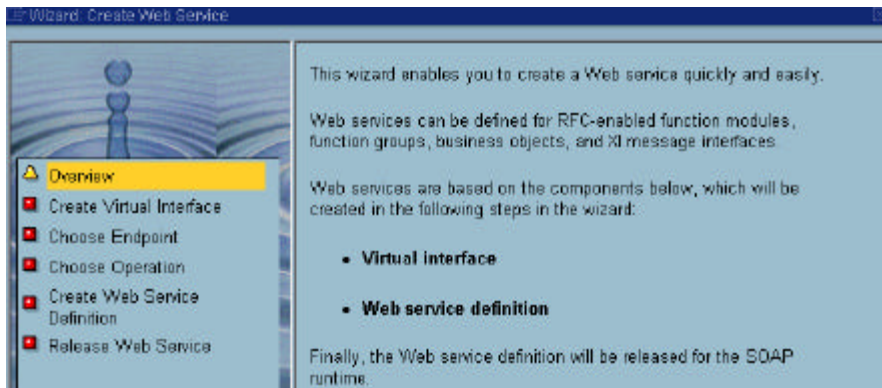
- **BAPI_BUPA_ADDRESS_CHANGE**
Brinda la posibilidad de cambiar la dirección relevante del BP. Se chequea la existencia del BP y de la dirección así como la consistencia de los nuevos datos. La tabla RETURN devuelve eventuales errores. En caso de error la dirección no viene alterada.
- **BAPI_BUPA_ADDRESS_GETDETAIL**
Devuelve información sobre la dirección y la comunicación del BP. La estructura ADDRESSDATA devuelve la dirección del BP. El parámetro RETURN devuelve los eventuales errores.
- **BAPI_BUPA_BANKDETAILS_GET**
Lee toda la información bancaria del BP y la devuelve en la tabla BANKDETAILS. Se chequea la existencia del Business Partner. Si se ingresa una fecha válida, entonces la función retorna la información bancaria válida para esa fecha. Si se desea obtener toda la información bancaria para ese BP, independientemente de su validez, entonces el parámetro VALID_DATE debe ser blanco.
- **BAPI_BUPA_BANKDETAIL_CHANGE**
Por medio de esta función es posible cambiar la información bancaria de un determinado BP. El sistema chequea la existencia del BP y de la información bancaria y luego verifica que la información ingresada sea válida. Los eventuales errores se devuelven en la tabla Return.
- **BAPI_BUPA_PCARD_CHANGE**
Permite modificar la información de tarjetas de crédito del BP.
- **BAPI_BUPA_PCARD_GETDETAILS**
Devuelve toda la información sobre las tarjetas de crédito del BP en la tabla de salida DATA. Si el BP no existe o no tiene tarjetas de crédito asignadas, se devuelve el error correspondiente en la tabla RETURN.

a) Pasos para la creación de un ABAP Web Service por medio del Wizard

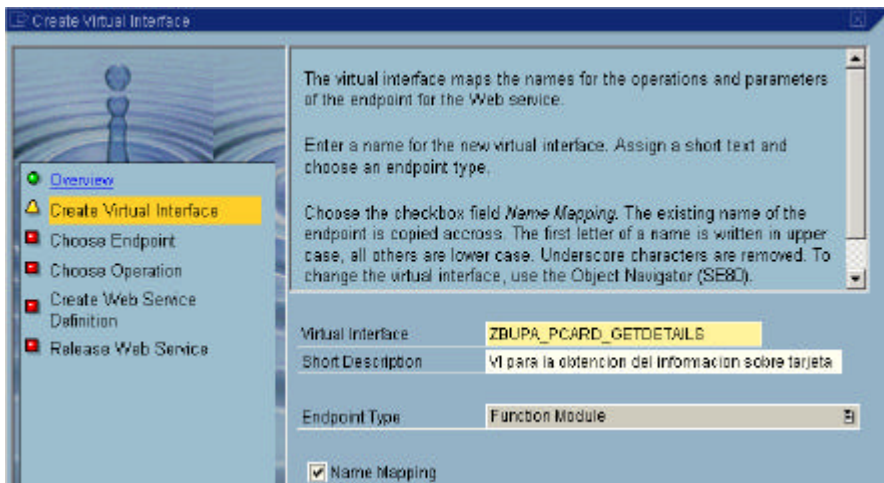
1. Desde la transacción SE37 ingresar el nombre BAPI_BUPA_PCARD_GETDETAILS y presionar DISPLAY.
2. Seleccionar la opción del menú *Utilities ->More Utilities -> Create Web Service -> From de Function Module* (como se muestra en la siguiente figura).



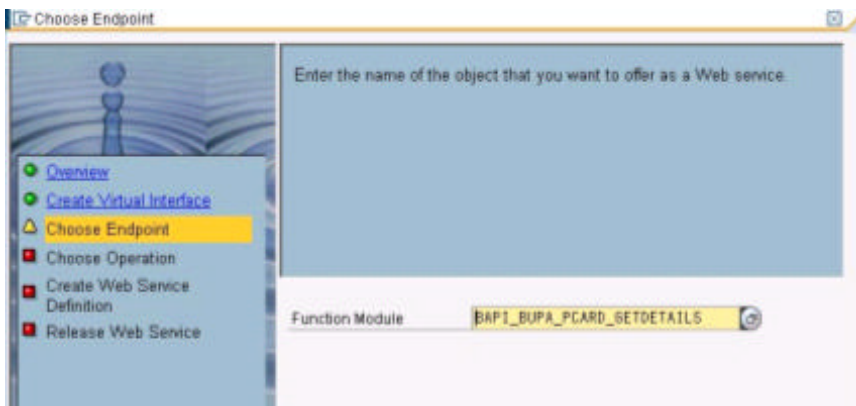
3. De esta forma viene lanzada la transacción WS_WZD_START que inicia el wizard para la creación del Web Service ABAP. Pulsar CONTINUE.



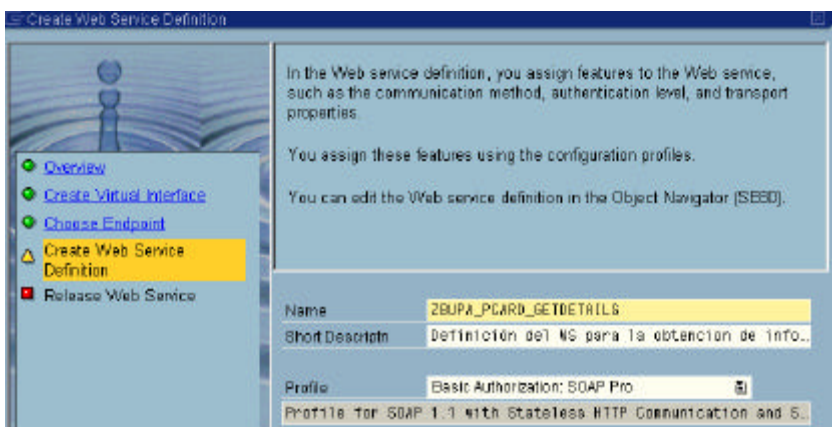
4. En este paso se crea la Interfaz Virtual ZBUPA_PCARD_GETDETAILS, introduciendo además una breve descripción y seleccionando la casilla "Name Mapping". Pulsar CONTINUE.



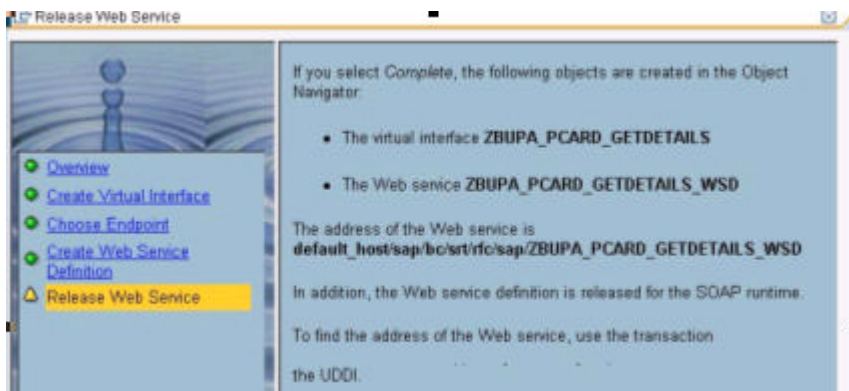
- Ingresar el nombre de la BAPI en la que se basa el Web Service:
BAPI_BUPA_PCARD_GETDETAILS.



- En este punto se crea la Definición del Web Service llamada ZBUPA_PCARD_GETDETAILS_WSD. Se ingresa una breve descripción y luego presionar CONTINUE.



7. La última pantalla del wizard nos indica si deseamos completar la creación del Web Service y luego presiona CONTINUE.



8. Finalmente, grabamos la Interfaz Virtual y la Definición del Web Service como objetos locales.

Con pocos pasos hemos creado un ABAP Web Service basado en la BAPI: **BAPI_BUPA_PCARD_GETDETAILS**.

Los nuevos objetos creados se pueden ubicarlos desde la transacción SE80 (“Objetos Locales”), bajo la jerarquía: *Enterprise Services -> Web Service Library*

b) Pasos para la creación manual de un ABAP Web Service

A continuación se creará un Web Service ABAP de manera **manual**, es decir **sin utilizar el wizard** como se hizo en el paso anterior (a). Para ésto, se debe crear lo siguiente:

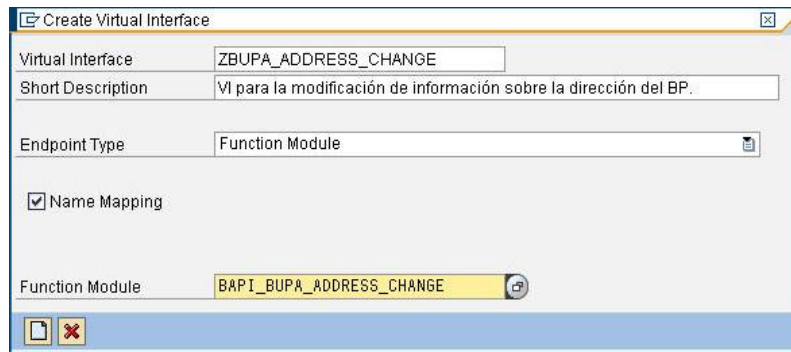
- Interfaz Virtual
- Definición del Web Service
- Configuración del Web Service

Creación de la Interfaz Virtual

Una Interfaz Virtual es la representación visual de un Web Service al exterior. De esta forma, es posible definir múltiples vistas de un Web Service y publicarlas separadamente.

1. Posicionarse en el **Object Navigator (SE80)**
2. Desplegar el menú contextual de “Enterprise Services” de los Objetos Locales.
3. Seleccionar *Create -> Virtual Interface*

4. Ingresar el nombre ZBUPA_ADDRESS_CHANGE, una breve descripción, seleccionar la casilla “Name Mapping” y como Function Module ingresar B API_BUPA_ADDRESS_CHANGE.



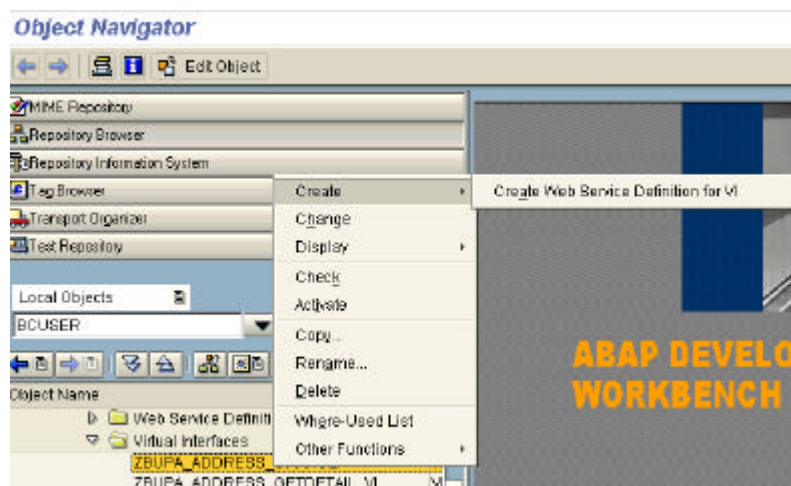
5. Pulsar SAVE, grabar el objeto como local y presionar la tecla para Activarlo.

Creación de la Definición del Web Service

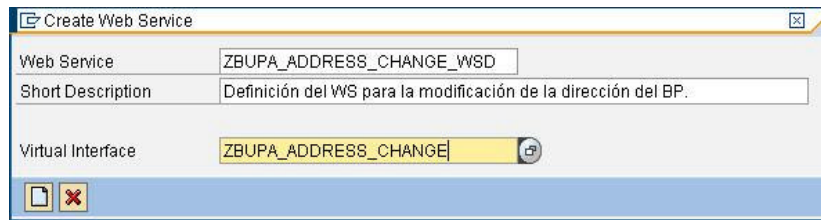
Una vez creada la Interfaz Virtual es posible crear múltiples definiciones del Web Service. En una definición se establecen características tales como el tipo de comunicación o el nivel de autenticación en forma abstracta. Posteriormente, en la configuración del Web Service, se definen los detalles técnicos.

Para crear la definición de un Web Service realizamos los siguientes pasos:

1. Posicionarse en el Object Navigator (SE80)
2. En Objetos Locales, dentro de *Enterprise Services -> Web Service Library -> Virtual Interfaces* seleccionar ZBUPA_ADDRESS_CHANGE y desplegar el menú contextual.
3. Seleccionar la opción *Create -> Create Web Service Definition for VI*



4. A continuación ingresar el nombre de la definición ZBUPA_ADDRESS_CHANGE y una breve descripción de la misma, luego pulsar el botón SAVE y guardar el objeto como local.



5. Por último, necesitamos activar la definición que se ha creado.

Características de la Definición del Web Service

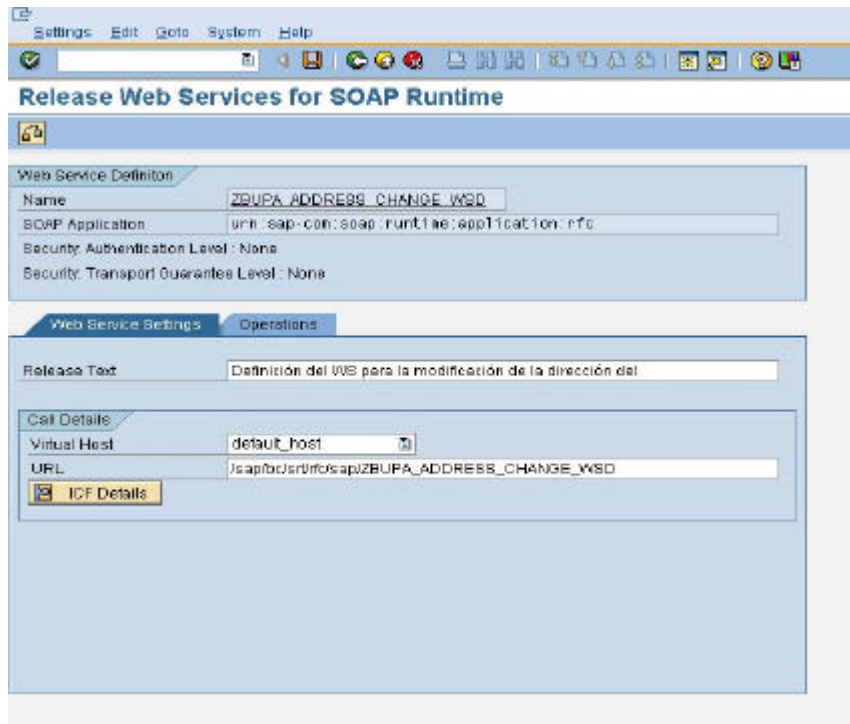
Si seleccionamos la definición ZBUPA_ADDRESS_CHANGE bajo el nodo *Web Service Definitions* podemos modificar las siguientes opciones:

- **Session-Oriented Communication:** En caso que la comunicación sea *stateful* debemos seleccionar la opción “Select Feature”
- **Authentication:** La opción preseada es “No Authentication”. Podemos seleccionar “Basic (user/password)” si la llamada al Web Service del lado del cliente requiere autenticación por medio de usuario y password. Se debe seleccionar la opción “Strong”, si la autenticación se llevará a cabo usando el protocolo SSL.
- **Transport Guarantee:** Debemos seleccionar las opciones “Integrity” o “Confidentiality” si los datos pasados desde los clientes del Web Service al servidor por medio de SOAP deben ser encriptados. En tal caso se utiliza SSL.

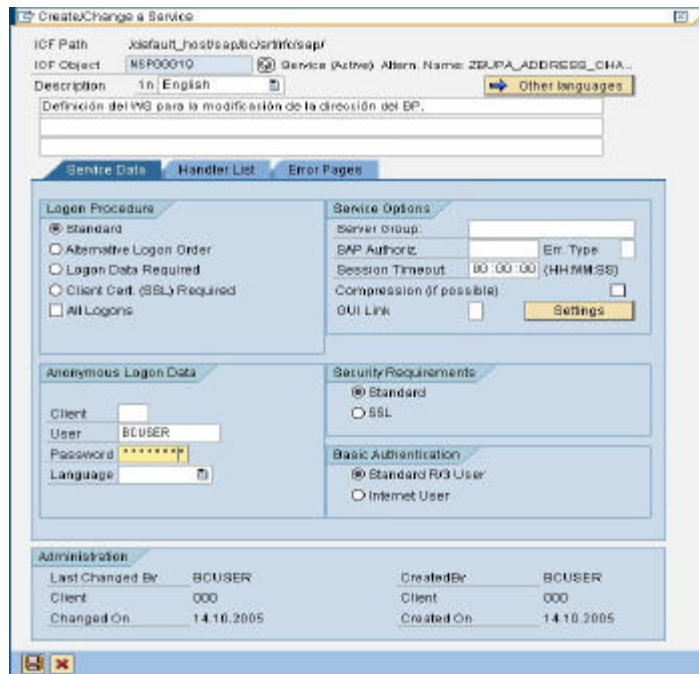
Configuración del Web Service

La creación de la configuración del Web Service se realiza por medio de la transacción “*Release Web Services for SOAP Runtime*” (**WSCONFIG**)

1. Desde la transacción **WSCONFIG** ingresamos el nombre de la definición ZBUPA_ADDRESS_CHANGE_WSD y presionamos el botón CREATE.
2. Podemos dejar el Virtual Host y URL propuesto, presionar SAVE y grabar como objeto local.



3. En este paso, presionar el botón “ICF Details”, y en la ventana “Maintain Service” hacer *doble click* sobre ZBUPA_ADDRESS_CHANGE_WSD.
4. Se presenta la ventana para cambiar la configuración del Web Service. En esta pantalla se puede especificar el comportamiento del logon, como así también los requerimientos de seguridad del Web Service. (Algunas consideraciones de seguridad de los Web Services, más adelante en la sección III). Para nuestro ejemplo, oprimir el botón CHANGE e ingresar el usuario *BCUSER* y la password *MINISAP*.



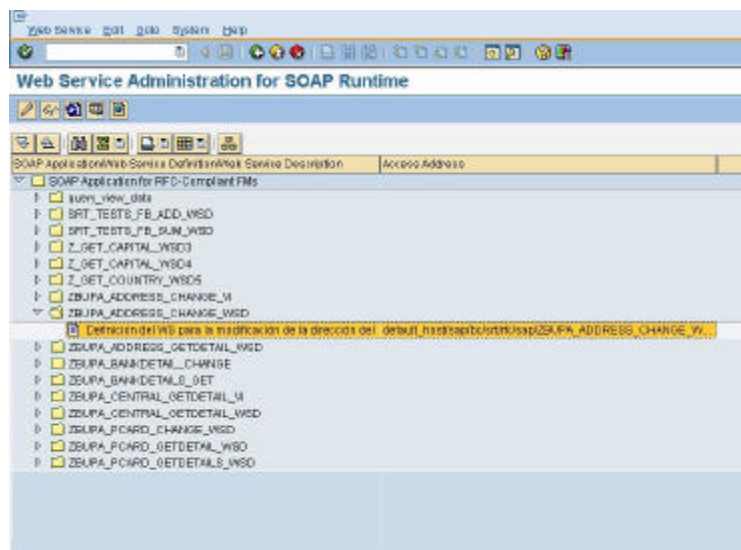
5. Luego, presionar STORE. Si aparece un mensaje de *warning*, ignorarlo, y posteriormente cerrar la ventana de configuración. De esta forma queda completada la configuración del Web Service.

Pasos para consumir el ABAP Web Service

Los Web Services que hemos creado ya sea a través del wizard o de forma manual, pueden ser “consumidos” por múltiples aplicaciones. En caso de contar con un motor J2EE en el servidor de aplicación podemos acceder a la homepage del Web Service, obteniendo información sobre la funcionalidad que brinda y testeando el servicio.

Para ello se necesitan realizar los siguientes pasos:

1. Lanzar la transacción **WSADMIN**, la cual lista todos los Web Services que han sido publicados en el SOAP runtime.
2. Bajo “SOAP Application for RFC-compliant FMs” desplegar el nodo ZBUPA_ADDRESS_CHANGE.
3. Seleccionar la configuración creada en los pasos previos y oprimir el botón “**Web Service Homepage**” (**CTRL + F8**)



4. En la ventana “*Settings for WSDL Generation*” seleccionar la opción “Include SAP Features in WSDL” y presionar OK.
5. A continuación, se abre una instancia del browser donde debemos ingresar el usuario y password previamente definidos al configurar el Web Service, en nuestro caso, usuario *BCUSER* y password *MINISAP*.

III. Consideraciones sobre seguridad de Web Services

Una de las principales razones por las que las empresas adoptan la tecnología de web services es porque permite establecer una comunicación abierta entre aplicaciones, independientemente de la plataforma. Sin embargo, estos beneficios implican tener en cuenta algunas consideraciones sobre aspectos de la seguridad en la implementación de Web Services. En este sentido, va a ser importante que sólo los usuarios con las autorizaciones adecuadas puedan consumir y ejecutar Web Services.

Existen tres puntos claves para implementar web services seguros:

1. Integridad de datos y confidencialidad
2. Autenticación
3. Autorizaciones

Desde el ABAP Workbench, en el momento de **crear la definición del web service** se puede elegir entre dos alternativas:

- Basic Authorization Profile: se utilizará identificación de usuario y password. Esta opción es adecuada para servicios donde la seguridad no es un punto crítico.
- Secure SOAP Profile: significa que se utilizarán para la autenticación certificados X.509

En la sección de **Autenticación** se puede seleccionar el nivel de autenticación entre:

- Strong: utiliza certificados X.509
- Basic: fuerza el uso de identificación de usuario y de password.
- None: se aplica cuando no necesitamos autenticación y deseamos exponer el web service a todo aquél que desee consumirlo.

Si se desea seleccionar el nivel de seguridad a nivel de transporte, se hará en la sección **“Transport Guarantee”** de la siguiente forma:

- None: transporta el web service como texto plano a través del protocolo HTTP
- Integrity: valida el web service contra un “message digest” para asegurar que no haya sido dañado durante el transporte.
- Confidentiality: utiliza SSL para encriptar la comunicación.
- Both: es una combinación de Integrity y Confidentiality.

En párrafos previos se mencionó que los usuarios deben tener las autorizaciones adecuadas para ejecutar web services. Estas autorizaciones deben ser combinadas con el rol (transacción PFCG) y asignadas al usuario. Para iniciar el web service el usuario debe tener el objeto de autorización S_SERVICE asignado a través de un rol de autorización.

Para que quede registrado el usuario, la fecha y hora de ejecución de un Web Service, en el momento de definir un web service se puede seleccionar la opción **“Trace Settings”**. Estas anotaciones pueden visualizarse luego en la transacción SM59 (RFC -> Display Trace). Si se desean visualizar las entradas en el log del sistema por Web Services cancelados, se usa la transacción SM21.

Con los recursos de seguridad mencionados en párrafos anteriores, SAP ofrece a sus clientes la posibilidad de implementar sólidas políticas de seguridad para evitar los probables riesgos cuando se adopta la tecnología de Web Services.

IV. Dónde obtener información adicional

SAP Developers Network www.sdn.sap.com

SAP NetWeaver - <http://www.sap.com/solutions/netweaver/>

IMPORTANTE

Copyright Diciembre 2005 Teknoda S.A. 2005. SAP, R/3 y ABAP son marcas registradas de SAP AG. Teknoda agradece el permiso de SAP para usar sus marcas en esta publicación.

SAP no es el editor de esta publicación y no es, por lo tanto, responsable de su contenido.

La información contenida en este artículo ha sido recolectada en la tarea cotidiana por nuestros especialistas a partir de fuentes consideradas confiables. No obstante, por la posibilidad de error humano, mecánico, cambios de versión u otro, Teknoda no garantiza la exactitud o completud de la información aquí volcada.

Dudas o consultas: sapping@teknoda.com