

## Notas técnicas de SAP / ABAP - Tip en detalle Nro. 11

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

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

#### ”2) El modelo BSP: Cómo construir las primeras BSP”

- Tema:** Netweaver, SAP Web AS, Business Server Pages, JAVA, Internet .
- Descripción:** El objetivo de la serie de tips que comenzamos a publicar ahora y por los próximos meses es **recorrer y ejemplificar el nuevo escenario de desarrollo que SAP ofrece a partir de Netweaver**. En este tip, explicamos paso a paso cómo construir las primeras BSP sencillas.
- Nivel:** Intermedio.
- Versión:** 6.1, 6.2, 6.3
- Fecha pub:** Octubre 2003
- Próximos Tips:** El modelo BSP (II): Un caso práctico con JavaScript  
El modelo BSP (III): Explotando BAPI's desde una BSP.  
El modelo BSP (IV): Las extensiones BSP y el Model View Controller SAP como JAVA Server. Netweaver Development Studio.  
Integración de JAVA y SAP a través del Java Connector.  
Entendiendo las Web Dynpro. Cómo construir una Web Dynpro.  
Web Services

**NOTA:** Para entender este tip es necesario haber leído el tip anterior de la serie “**Entendiendo el SAP Web Application Server desde el punto de vista del programador**”. Consulte nuestro sitio [www.teknoda.com/tipssap.htm](http://www.teknoda.com/tipssap.htm) para accederlo

*"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](mailto:sapping@teknoda.com), indicando su nombre, empresa a la que pertenece, cargo y país.*

---

### Tabla de contenido

- I. Introducción General (repasso del tip precedente)
- II. Introducción a las BSP
- III. Estructura de una BSP Application
- IV. Creación de una BSP Application con una página estática
- V. Creación de una BSP Application con una página dinámica que despliega datos de una tabla.
- VI. Para tener en cuenta. Conceptos adicionales

---

## I. Introducción / Resumen Ejecutivo

El objetivo de la serie de tips que comenzamos a publicar ahora y por los próximos meses es **recorrer y ejemplificar el nuevo escenario de desarrollo que SAP ofrece a partir de Netweaver**.

Netweaver no es el nombre de un producto. SAP Netweaver es la definición de un **“framework”**, un marco o infraestructura tecnológica a la que conforman todas las soluciones SAP existentes y futuras. El componente basal de la infraestructura SAP Netweaver es el **SAP Web Application Server (SAP Web AS)**.

A partir de la versión 6.1, **SAP desplaza el SAP BASIS System y lo reemplaza con el SAP Web Application Server**. Concebido para actuar en un mundo dominado por las aplicaciones “Web enabled”, el SAP Web AS es la respuesta de SAP para las demandas de escenarios e-business. Basado en el mismo modelo de tres capas (three-tier) del conocido SAP Application Server, el SAP Web AS **trae la funcionalidad Web completamente embebida**, tanto desde el punto de vista de los protocolos de comunicaciones como del servicio de aplicaciones. El SAP Web AS incluye soporte NATIVO para los protocolos de Internet, eliminando la necesidad de un Internet Gateway separado. Cubre la función de cualquier Web Server en el sentido de que es capaz de entender HTML, XML, WML y SMTP nativamente. Como Web Application Server, es capaz de entender ABAP, y también JAVA.

Se dice a menudo que el SAP Web AS es una plataforma de “doble personalidad” porque funciona como un Kernel para SAP R/3, brindando servicios como Web Application Server “nativo”, y a la vez, es capaz de actuar como un servidor de aplicaciones JAVA, conformando enteramente con el estándar J2EE.

### SAP Web AS como R/3 Kernel y Web Application Server Nativo

El SAP Web AS, al igual que el BASIS que lo precede, **provee el cimiento tecnológico sobre el cual se apoya el R/3 Enterprise**, el sucesor de R/3, y la mayoría de los componentes de mySAP.com. En este rol, el SAP Web AS funciona como el R/3 Kernel, definiendo toda la arquitectura de conectividad, seguridad, acceso a bases de datos, y también la interfaz de programación.

Como Application Server Nativo, el SAP Web AS continúa soportando el universo ABAP de versiones anteriores, pero introduce nuevas herramientas para desarrollo de aplicaciones Web fuertemente integradas a R/3. Desde la primera versión del Web AS, (6.1) aparece el modelo de **Business Server Pages (BSP's)** como una potente opción para construir aplicaciones Web que accedan dinámicamente a los datos de SAP.

Las BSP son en esencia páginas Web, compuestas por el código HTML estático, más una porción de “scripts” escritos en ABAP (o JAVAScript). Implementan un esquema muy parecido al de las Java Server Pages (JSP's) y las Active Server Pages (ASP's), ambas ampliamente usadas en el desarrollo de aplicaciones para la Web. El código de los scripts es interpretado por el servidor y proporciona los contenidos dinámicos (por ejemplo, el contenido de un tabla), que luego son mergeados con el resto de la página. En este tip cubriremos una explicación del modelo de BSP's, y dos ejemplos sencillos de las mismas.

---

## II. Introducción a las BSP

Las Business Server Pages (BSP's) son una de las formas en que pueden definirse páginas Web dentro del Web Application Server SAP. Consisten en código HTML que define los componentes estáticos de la página, con código ABAP embebido para gestionar los elementos dinámicos, tomados de R/3 u otras fuentes. Las BSP's **pertenecen al universo de desarrollo “nativo” del SAP Web AS, es decir, su personalidad ABAP**.

Arquitecturalmente, las BSP's implementan un esquema muy parecido al de las Java Server Pages (JSP's) y las Active Server Pages (ASP's), ambas ampliamente usadas en el desarrollo de aplicaciones para la Web. Al código embebido en lenguajes ajenos al HTML se lo denomina “script”, y cuando este código es resuelto por el servidor (no por el browser), hablamos de “Server-side scripting”.

Cuando una BSP es apuntada por una URL, el Web Application Server se encarga de “parsear” (dividir) los elementos estáticos de los dinámicos. El código de los “scripts” es interpretado por el servidor y proporciona los contenidos dinámicos (por ejemplo, el contenido de un tabla), que luego son “mergeados” con el resto de

la página. El browser siempre recibe el código HTML resuelto, en una forma en que lo puede representar directamente.

Las BSP's soportan también la opción de "server-side scripting" en JavaScript. El uso de JavaScript en este contexto es simplemente un tema de preferencias sintácticas, dado que tanto en ABAP o en JavaScript, las estructuras de datos subyacentes son variables ABAP (campos, estructuras, tablas internas). **No debe confundirse el JavaScripting de las BSP's con la personalidad JAVA del Web AS.**

Lo más interesante de las BSP's en términos de la comunidad SAP es que puede definirse la lógica de una aplicación para la Web usando el lenguaje y las estructuras ya conocidas de ABAP. Cualquier programador ABAP debería poder construir BSP's con un mínimo de entrenamiento.

A diferencia de las restricciones que planteaba un entorno como el ITS, las BSP's permiten un modelo de desarrollo flexible.

Para construir y testear las BSP's, se trabaja con un nuevo componente del ABAP Workbench denominado Web Application Builder, accesible desde la transacción SE80. El Web Application Builder es una colección integrada de herramientas que incluye, entre otras cosas, un BSP editor y un debugger para los scripts ABAP y JAVA

Para definir una BSP es necesario definir primero una BSP Application que la contenga. Una BSP Application es un conjunto de páginas HTML. Tanto las BSP's como las BSP Applications forman parte del repositorio de SAP.

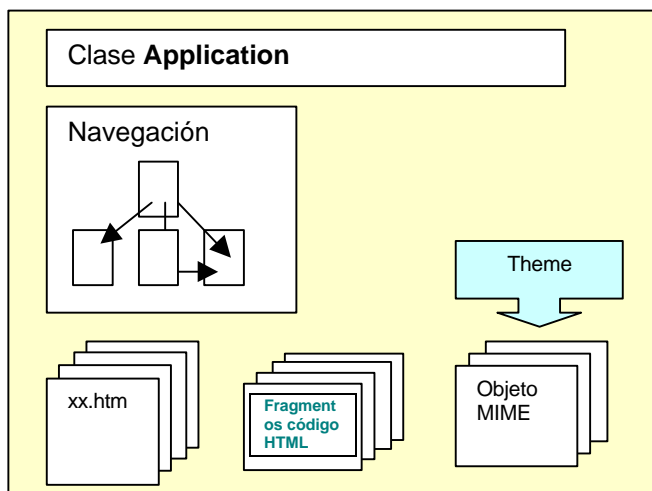
---

### III. Estructura de una BSP Application

Una BSP Application (Fig. 1) es una aplicación Web autosuficiente, con su funcionalidad de presentación, navegación y lógica de la aplicación embebida. Está integrada por una o más páginas BSP, con su código HTML y scripts escritos en JavaScript o ABAP.

La BSP brinda la posibilidad de acceder directamente al corazón de SAP "en lenguaje ABAP" a través de lo que se denominan eventos. Los eventos son puntos fijos y predeterminados en el ciclo de vida de una página web. En estos eventos se pueden utilizar funciones, BAPIs y ABAP OBJECTS.

Fig. 1 – Estructura de una BSP Application



Las BSP's son parte de una BSP Application y usualmente tienen extensión **.htm** . Los siguientes son los componentes que integran una BSP (Fig. 2)

- Attributes
- Layout
- Events Handlers
- Page attributes
- Type definitions
- Preview

**Attributes:** como su nombre lo indica, se especifican los atributos que la página posee, con la ventaja de que algunos ya vienen establecidos por default, como por ejemplo el protocolo y la URL. También se especifica si la página es de tipo *page* o *page fragment*.

Si se define una página de tipo *page fragment*, esta podrá ser incluida en todas las páginas que se desee. Dentro de las *page fragment* se describen las características comunes de las páginas de la BSP Application tales como: fuente, tamaño de la fuente, color de fondo de la página , etc.

**Layout:** se define cómo aparecerá la página en el browser. Aquí es donde se describen los elementos estáticos en html (aquellos que darán el formato a la página ) y también los elementos dinámicos . Es importante recordar que los elementos dinámicos pueden ser escritos indistintamente tanto en JavaScript como en ABAP.

**Events Handlers:** son eventos predeterminados que permiten el intercambio de información con el sistema R/3. Debido a que definen el ciclo de vida de una página web, se deben detallar en un orden específico. Como restricción, deben estar escritos exclusivamente en lenguaje ABAP. Es una de las componentes principales de la estructura de una BSP debido a la facilidad provista para la comunicación dinámica con el sistema R/3.

La siguiente lista muestra cada uno de los eventos predeterminados, junto con sus características y usos más frecuentes:

- **On Create:** es el primer evento en ejecutarse. Es utilizado para la inicialización de datos y la instanciación de objetos.
- **On Request:** es ejecutado cada vez que la BSP es invocada. Es el evento encargado de recuperar la estructura de datos internos de acuerdo al requerimiento solicitado por el usuario.
- **On Initialization:** este evento es utilizado para la recuperación de datos (realiza lecturas de la base de datos y completa tablas internas).
- **On Manipulation:** no es usado comúnmente y sirve para manipular una cadena de datos HTTP después que la página ya ha sido desplegada.
- **On Input Processing:** es utilizado principalmente para la validación y procesamiento de los datos ingresados por el usuario (por ejemplo: el servidor puede validar los datos del usuario contra una base de datos) . Además puede utilizarse para indicar la próxima página que continúa en el mapa de navegación.

**Page Attributes:** en esta sección los atributos de la página se definen como campos de tablas , work areas, tablas internas o simplemente una cadena de caracteres. Estos atributos de páginas pueden ser pasados como parámetros a otra página, dentro de la misma BSP Application, al realizar la navegación sobre la página; o pueden ser usados como variables globales dentro de la BSP. En este último caso, tanto los Events Handlers como el Layout pueden “ver” y utilizar estas variables.

**Type Definitions:** se definen los tipos (types) de ABAP que van a ser accedidos por algún evento utilizado dentro de la página. De la misma manera, se define un tipo perteneciente a una tabla interna , el que posteriormente se asignará como tipo de un elemento de la componente *Page Attributes*.

**Preview:** en esta sección podemos ver cuál es el aspecto de los elementos estáticos que aparecerán en el browser cuando la BSP sea ejecutada.

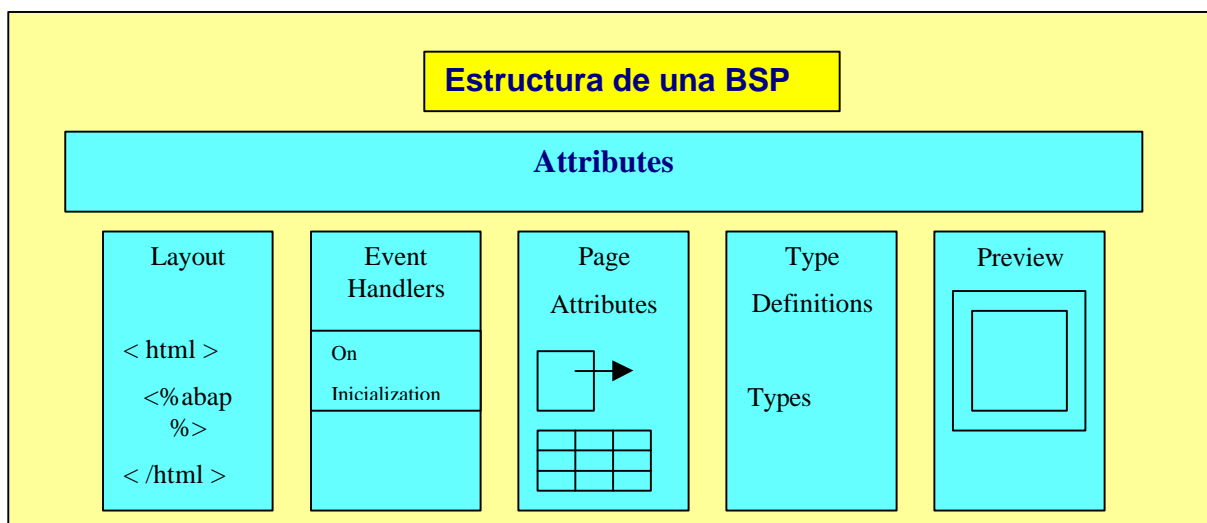


Fig. 2 Estructura de una BSP

Las componentes de una BSP mencionadas anteriormente serán específicas de manera sencilla y amigable por medio de la herramienta **Web Application Builder** al confeccionar una BSP. Los pasos a seguir se indican en la próxima sección.

#### IV. Creación de una BSP Application con una página estática


En la presente sección, se mostrarán los pasos necesarios para crear una BSP simple, que sólo contendrá un texto estático (“Hola a todos”). Esta BSP no incluye la definición de los siguientes items: scripts, Page Attributes, Event Handlers.

Para crear y editar BSP’s, se utiliza el **Web Application Builder**. Esta herramienta pertenece al ABAP Workbench y se puede acceder directamente desde **Object Navigator** (transacción SE80).

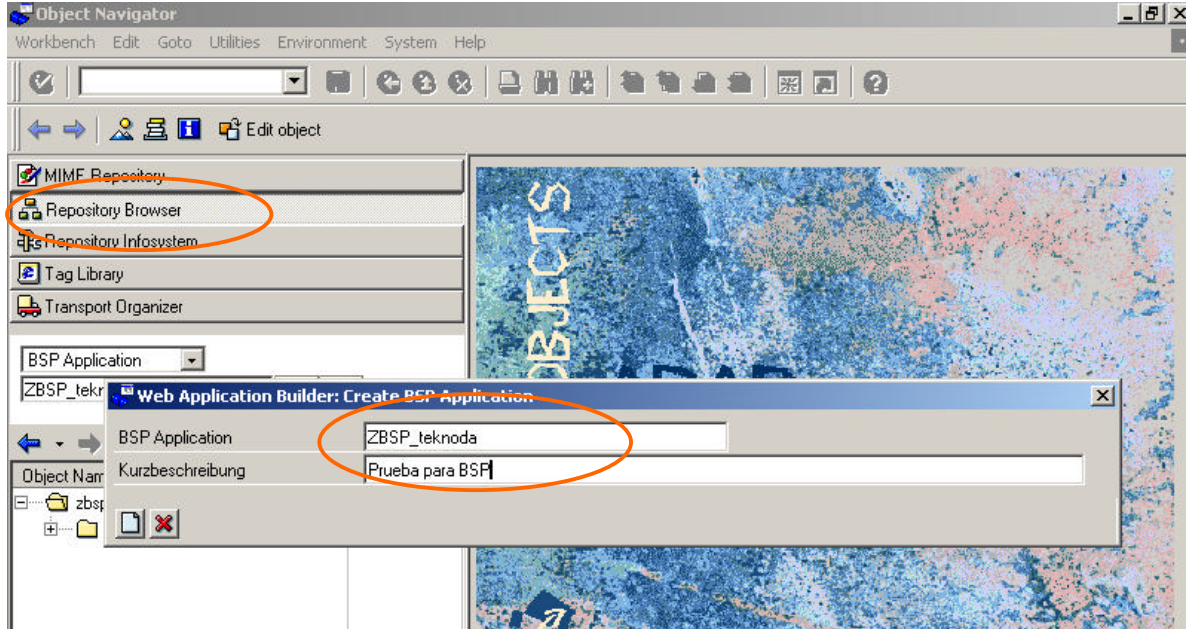
Como mencionáramos anteriormente, es importante recordar que las BSP’s deben ser siempre parte de una BSP Application. Por lo tanto, antes de construir una BSP, se debe crear una **BSP Application** o de lo contrario utilizar una **BSP Application** ya creada.


Los siguientes pasos detallan el proceso de creación de una BSP, comenzando desde la generación de la BSP Application que la contendrá:

Ingresar al **Object Navigator** y realizar un click sobre la opción “Repository Browser”.

1. Desde el menú desplegable del *Repository Browser*, seleccionar **BSP Application**.
2. En la caja de texto, ingresar el nombre de la nueva BSP Application y realizar un click en el botón de **display** (  ). En el ejemplo la aplicación se denominará “ZBSP\_Teknoda”.
3. Si la aplicación no existe, aparece la ventana **Create Object**, indicando que la aplicación no existe y preguntando si se desea crearla. Clickear **Yes**. Al aceptar la creación de la BSP Application, se despliega la ventana **Web Application Builder: Create BSP Application** en donde se debe agregar

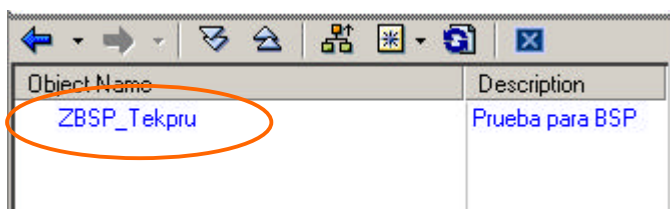
una breve descripción. Luego presionar Enter. La siguiente imagen muestra la creación de la BSP Application:



Una vez creada la BSP Application, debe ser activada (  o CTRL +F3).

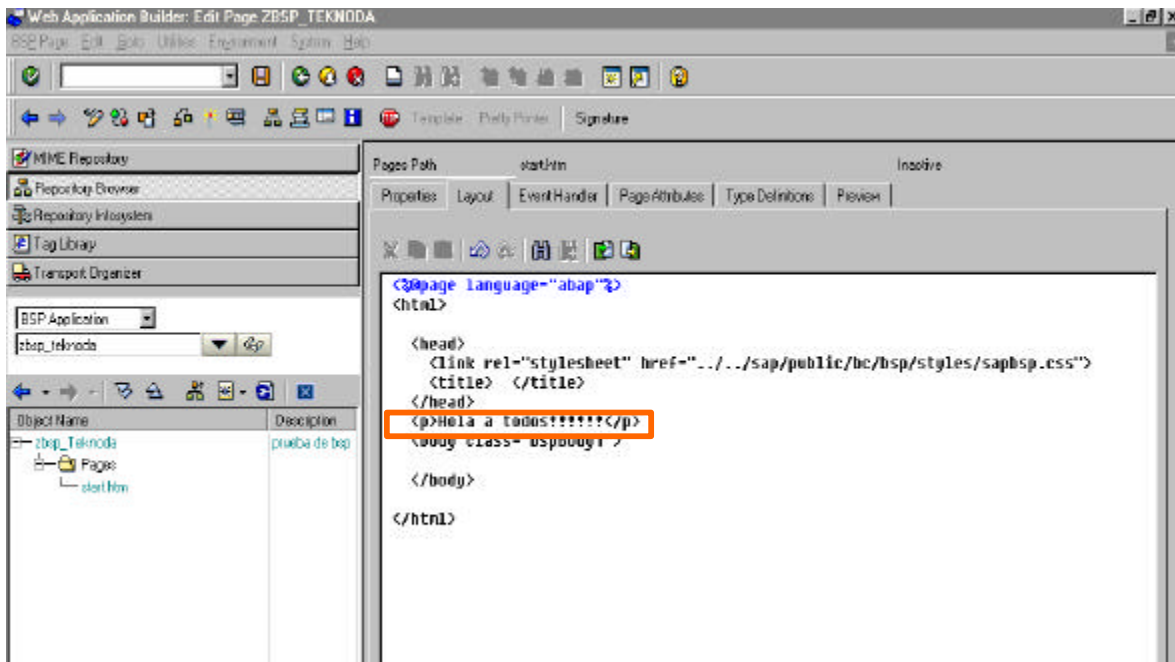
En el caso que la aplicación exista, aparecerá, en la esquina inferior izquierda, un árbol con la estructura de la aplicación y la vista del editor. Luego se podrá navegar por las páginas existentes o crear otras nuevas.


4. En la siguiente ventana, **Create Object Directory Entry**, ingresar un nombre de paquete y salvar la entrada. Si la BSP Application es simplemente una prueba o ejemplo, salvarla como un Objeto Local y asignarle el paquete \$TMP. **En la parte inferior izquierda de la ventana del Object Navigator (siguiente imagen) puede visualizarse el “contenedor vacío” que se ha creado para la BSP Application. El próximo paso será generar una BSP.**

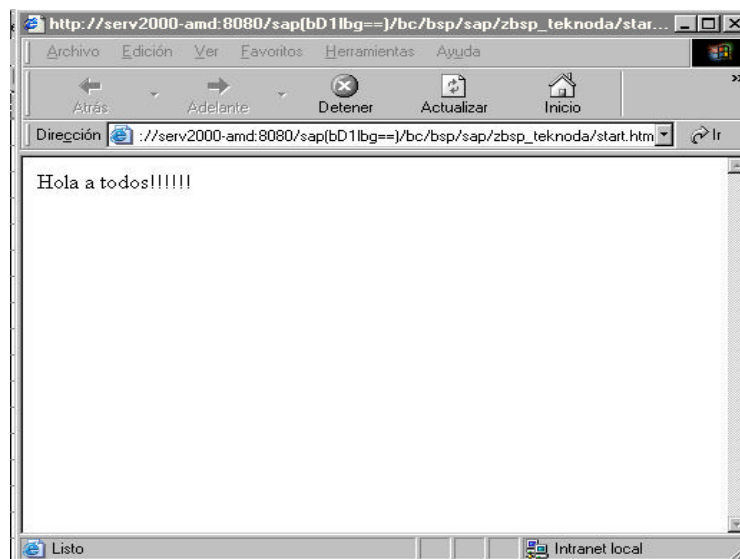


5. Para generar una BSP, ubicarse sobre la BSP Application recientemente creada, y presionar botón derecho del mouse. Desde el menú de contexto, seleccionar **Create** y luego **Page**. Aparece la ventana **Web Application Builder: Create Page**. Ingresar el nombre y una breve descripción para la página que se creará. El checkbox **Page Fragment** permite establecer si la página a crearse es una porción de código HTML donde se define el formato de las páginas de la BSP Application. Ejemplo: cabecera.htm. Para crear, presionar Enter. A la derecha de la ventana del Object Navigator, se puede visualizar el Layout básico de la página recientemente creada.

- Ahora podemos ingresar al código fuente que se encuentra en el layout, en el editor del frame, y realizar una pequeña modificación:



- Luego se debe activar la BSP Application junto con la página start. htm creada.
- Para ejecutar la BSP Application, presionar  o F8. Se despliega automáticamente la página en el Web Browser, invocada a través de la URL que está definida en la componente *properties* de la página start.htm.



Se puede también probar que la BSP Application no depende exclusivamente del *server* SAP. Esto se puede verificar copiando la URL que aparece en el Web Browser y levantando la BSP Application desde cualquier otro Web Browser.

## V. Creación de una **BSP Application** con una página dinámica que despliega datos de una tabla de un sistema **SAP R/3**

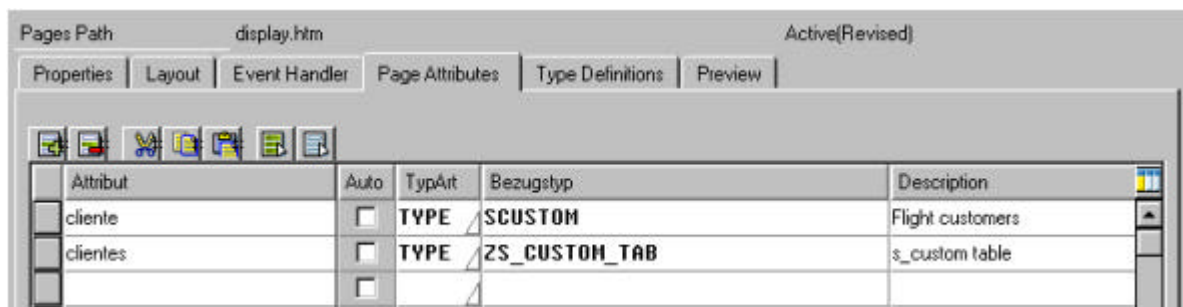
En el siguiente ejemplo , se muestra como se despliega en un Web Browser el contenido de una tabla de un sistema SAP R/3. El objetivo es recuperar y usar datos en una BSP, usando ABAP. En la siguiente BSP se trabaja con datos de clientes de la tabla *SCUSTOM*. La BSP a crear trabajará con sólo 3 componentes: el *Layout*, el *Page Attribute* (almacena los datos de los clientes) y el *Event Handler* (para realizar la consulta y poder obtener los datos a desplegar).

Para empezar, crear una BSP Application con nombre “**z\_bsp\_desplegar\_scustom**” y luego una página con extension **.htm**. Activar la BSP Application y la nueva página. Recordar los pasos a seguir para la creación de la BSP Application, ya detallados en la sección anterior. Una vez completados, continuar con los siguientes pasos:

1. Seleccionar el tab **Page Attributes** en la ventana del editor.
2. Colocar en la columna de **Attribute** el nuevo parámetro llamado “*clientes*” (observar la “s” final).
3. Seleccionar en la columna de **TypArt** la opción “*TYPE*”.
4. Colocar en la columna de **Associated Type** el tipo de dato ABAP “*zs\_custom\_tab*”, el cual es una estructura de la tabla SCUSTOM. Esta estructura debe ser creada previamente en el data dictionary. El parámetro “*clientes*” será usado para almacenar la lista de clientes recuperada desde la tabla SCUSTOM durante el runtime.

Adición de un segundo parámetro desde **Page Attributes**:

5. Completar el segundo renglón con los siguientes datos: **Attribute:** “*cliente*” (sin “s”), **TypArt:** “*TYPE*” y **Associated Type:** “*SCUSTOM*” ( tipo de dato ABAP, es una tabla standard de SAP). La siguiente imagen muestra los dos atributos ya incorporados:



The screenshot shows the SAP Page Attributes editor window. The title bar indicates the page path is 'display.htm' and it is 'Active(Revised)'. The 'Page Attributes' tab is selected. Below the tab are several icons for editing. The main area contains a table with the following data:

Attribut	Auto	TypArt	Bezugstyp	Description
cliente	<input type="checkbox"/>	TYPE	SCUSTOM	Flight customers
clientes	<input type="checkbox"/>	TYPE	ZS_CUSTOM_TAB	s_custom table

6. Seleccionar el tab **Layout** en la ventana del editor.
7. Verificar la existencia de la directiva `< % @page language="abap"% >` como primera en el layout. Esta directiva indica que se trabaja con ABAP.
8. Incorporar en el layout el código HTML como se muestra en la siguiente imagen:



```

<%@page language="abap"%>
<html>
  <head>
    <title> Fligth connection </title>
  </head>

  <h2>Esta es una tabla extraida del R/3</h2>
  <body>

    <table border = 1>
      <tr>
        <th>id</th>
        <th>name</th>
        <th>form</th>
        <th>street</th>
        <th>city</th>
        <th>country</th>
        <th>region</th>

<%Loop at Cliente into Clientes. %>
      </tr>
      <td><% =clientes-id %></td>
      <td><% =clientes-name %></td>
      <td><% =clientes-form %></td>
      <td><% =clientes-street %></td>
      <td><% =clientes-city %></td>
      <td><% =clientes-country %></td>
      <td><% =clientes-region %></td>
<% EndLoop. %>

    </table>
  </body>

</html>

```

El esqueleto de tags en HTML para mostrar tablas es el siguiente:

```

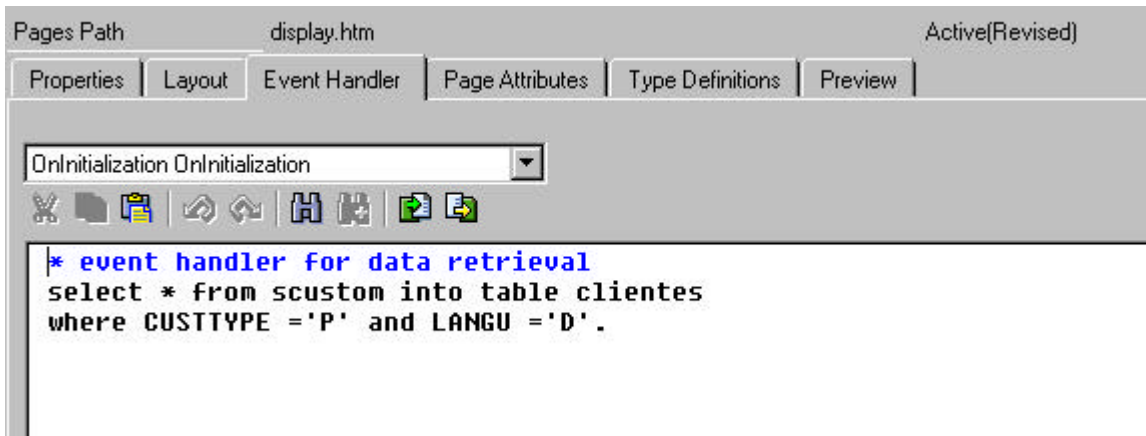
<table>
  <tr>
    <th></th>
    <th></th>
    <th></th>
    <th></th>
  </tr>
</table>

```

Recordar que los datos contenidos en la tabla interna “*clientes*” necesitan ser loopeados y mostrados registro por registro.

Adición de **Event handlers**:

9. Seleccionar el tab **Event Handlers**.
10. Seleccionar de la lista desplegable “**On Initialization**”.
11. Copiar la siguiente consulta :



12. Guardar las tareas realizadas.
13. Activar la BSP Application.
14. Ejecutar la BSP Application. Se desplegará automáticamente la página en el Web Browser:



Pueden seguirse los mismos pasos anteriores para crear una BSP que defina un formulario de entrada de datos. En este caso, el Event Handler a utilizar sería **OnInputProcessing**.

---

## V. Para tener en cuenta:

- Una BSP se compila dentro de una clase local de ABAP; por lo tanto, se puede depurar. Por ejemplo, es posible ver la carga de la estructura de la tabla.
- Cuando se activa la BSP Application, no se realiza el chequeo sintáctico del código HTML, ni de la codificación en JavaScript (si existe en el layout). Sólo controla código ABAP.
- No es posible alternar en un mismo layout codificación en JavaScript y en ABAP.
- Recordar que los eventos pueden escribirse sólo en ABAP.
- Las páginas pueden ser enriquecidas con imágenes y sonidos. Por ejemplo: `<%img src = "imagen.jpg" %>`.
- Los **page fragment** se utilizan para darle uniformidad en el diseño a la BSP Application.
- Las imágenes para la BSP Application pueden ser tomadas del **MIME Repository**. Los objetos MIME son almacenados físicamente en la base de datos del Web AS y son "servidas" en runtime.
- Es recomendable no utilizar la carpeta PUBLIC del MIME Repository. Se aconseja en cambio, generar una carpeta con el mismo nombre de la BSP Application, dónde se guardan las imágenes que ésta utilizará.

*Copyright 2003 Teknoda S.A. Octubre 2003. 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](mailto:sapping@teknoda.com)*