



SAP  
Records Management

## **Visualisierung von Records Management in BSP**

Dokumentation für Entwickler

17. Februar 2004

# Inhaltsverzeichnis

<b>1</b>	<b>EINLEITUNG</b>	<b>4</b>
<b>2</b>	<b>ENTWICKLUNG VON SERVICE PROVIDERN FÜR BSP</b>	<b>4</b>
<b>2.1</b>	<b>Programmiermodell</b>	<b>4</b>
2.1.1	BSP Client Klasse	5
2.1.2	BSP Query Klasse	5
2.1.3	BSP Controller Klasse	5
2.1.4	BSP View	6
2.1.5	Service Provider Backend	6
<b>2.2</b>	<b>Wichtige Schnittstellen</b>	<b>6</b>
2.2.1	BSP Client Klasse	6
2.2.2	BSP Query Klasse	7
2.2.3	BSP Controller Klasse	7
<b>3</b>	<b>VERWENDUNG DES BSP FRAMEWORK</b>	<b>9</b>
<b>3.1</b>	<b>Das &lt;srn:element&gt; Tag</b>	<b>9</b>
<b>3.2</b>	<b>Beispielprogramme</b>	<b>10</b>
<b>3.3</b>	<b>Attribute</b>	<b>10</b>
3.3.1	ID	10
3.3.2	EMPTY_CONTENT_TEXT	10
3.3.3	EMBEDDED	10
3.3.4	EVENT_CALLBACK	10
<b>4</b>	<b>AUSGELIEFERTE BSP SERVICE PROVIDER</b>	<b>10</b>
<b>4.1</b>	<b>Organizer</b>	<b>10</b>
<b>4.2</b>	<b>Akte</b>	<b>11</b>
<b>4.3</b>	<b>Dokumente</b>	<b>11</b>
<b>4.4</b>	<b>Archivdokumente</b>	<b>11</b>
<b>4.5</b>	<b>Notizen</b>	<b>11</b>
<b>4.6</b>	<b>URL</b>	<b>11</b>
<b>4.7</b>	<b>Suchdialog für Akten, Dokumente und Notizen</b>	<b>11</b>
<b>4.8</b>	<b>Implementieren einer Extension-Klasse</b>	<b>11</b>
4.8.1	Methode GET_ICON	11
4.8.2	Methode GET_PARA_BEE	11
4.8.3	Methode GET_PARA_TAB	11

4.8.4	Methode GET_RESULT_FIELD	12
4.8.5	Methode GET_TITLE	12
4.8.6	Methode RENDER_CELL	12
<b>4.9</b>	<b>Verwenden einer Extension-Klasse</b>	<b>12</b>
<b>5</b>	<b>VERWENDUNG DER BSP SERVICE PROVIDER IM PORTAL</b>	<b>12</b>

# 1 Einleitung

Für das Verständnis dieses Dokuments werden folgende Kenntnisse vorausgesetzt:

- Technischen Begriffe und Architektur des Records Management Framework. Informationen hierzu finden Sie in der Records Management Referenzdokumentation für Entwickler.
- Grundlagen von Business Server Pages und des Portals.

Das Dokument besteht aus vier Teilen, die z.T. für verschiedene Zielgruppen von Interesse sind:

- **Informationen zur Entwicklung von Service Providern**  
Diese Informationen sind primär für Entwickler interessant, die eine BSP Visualisierung für eine in selbst entwickelten Service Provider implementieren wollen
- **Informationen über die Verwendung des BSP Frameworks**  
Diese Informationen sind interessant für Berater und Entwickler, die die BSP Service Provider in eigenen Szenarien nutzen wollen, z.B. innerhalb selbst entwickelter BSP Applikationen oder im Portal.
- **Informationen zu den ausgelieferten Service Providern**  
Diese Informationen sind interessant für Berater und Entwickler, die die BSP Service Provider in eigenen Szenarien nutzen wollen, z.B. innerhalb selbst entwickelter BSP Applikationen oder im Portal.
- **Informationen zur Verwendung der BSP Service Provider im Portal**  
Diese Informationen richten sich speziell an Berater und Entwickler, die Records Management in einem Portalszenario nutzen wollen.

Dieses Dokument erklärt **nicht** die Grundlagen von Business Server Pages, des Portals oder von Records Management; hierfür sei auf die entsprechende Online-Dokumentation verwiesen.

## 2 Entwicklung von Service Providern für BSP

### 2.1 Programmiermodell

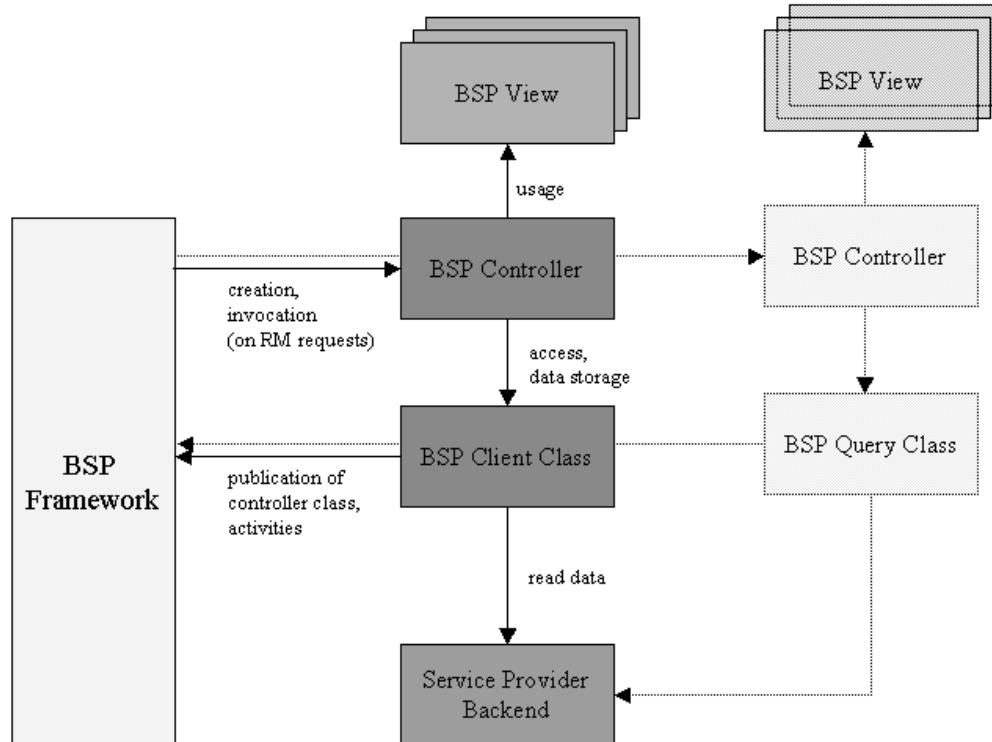
Das Programmiermodell für BSP -Service-Provider folgt dem Model -View-Controller (MVC) -Pattern; Voraussetzung für die Entwicklung von BSP -Service Providern ist ein grundlegendes Verständnis dieses Programmiermodells und der anderen Eigenschaften von Business Server Pages. (siehe Online-Dokumentation, Tutorial zu BSP).

Ein BSP-Service-Provider besteht aus vier Bausteinen:

- BSP Client Klasse
- BSP Query Klasse

- BSP Controller Klasse (mit BSP-Views)
- Service Provider Backend

Der Zusammenhang zwischen den Bausteinen wird in der folgenden Abbildung dargestellt:



### 2.1.1 BSP Client Klasse

Die BSP Client Klasse wird im Framework registriert. Sie dient der Aufbereitung der Daten aus dem Backend, sowie der lokalen Speicherung von Werten in einer Session.

Die Klasse muß die Klassenrolle `IS_SP_P_VISUALIZATION_BSP_CLASS` implementieren; sie wird in der RM Registry registriert.

Ebenfalls werden an der Client Klasse der Name des verwendeten BSP - Controllers publiziert sowie die angebotenen Aktivitäten und die Berechtigungsprüfung implementiert.

### 2.1.2 BSP Query Klasse

Analog zur BSP Client Klasse implementiert die BSP Query Klasse die Klassenrolle `IS_SP_VISUAL_QUERY_BSP_CLASS`; sie ist für die Suchaktivität erforderlich. Ihre sonstigen Eigenschaften entsprechen der BSP-Client-Klasse.

### 2.1.3 BSP Controller Klasse

Die BSP Controller Klasse wird über die BSP -Entwicklungsumgebung angelegt. Sie erbt gemäß den BSP -Vorgaben von `CL_BSP_???` und

implementiert das Interface `IF_SRM_SP_BSP_CONTROLLER`. Ihre Aufgabe ist das Behandeln von RM -Requests (Aktivitäten) und die Verarbeitung von Benutzerevents. Mit Hilfe der normalen BSP -Methoden ruft die Controller - Klasse einen oder mehrere Views für die Darstellung auf.

Die BSP Controller Klasse wird *nicht* in der RM Registry registriert. Da sie bei jedem http -Request neu erzeugt wird, ist sie *nicht* zur lokalen Speicherung von Daten zwischen zwei http-Requests geeignet.

#### 2.1.4 BSP View

Die BSP -Views dienen der Darstellung eines BSP -Service Providers. Es können beliebige HTML -Elemente sowie die Elemente der HTMLB und XHTMLB BSP-Extension genutzt werden.

*Wichtiger Hinweis: Da die BSP Views der Service Provider über das `<srmlb:element>` Tag in eine Seite eingebunden werden können, ist die Verwendung folgender HTMLB -Tags bzw. HTML -Elemente in BSP Views für Service Provider **verboten**:*

- `<htmlb:content>`
- `<htmlb:page>`
- `<htmlb:document>`
- `<htmlb:documentBody>`
- `<htmlb:documentHead>`
- `<HTML>`, `</HTML>`
- `<HEAD>`, `</HEAD>`
- `<BODY>`, `</BODY>`

#### 2.1.5 Service Provider Backend

Für die persistente Datenablage wird –wie in einem Service Provider für SAPGUI- das Backend in der Klassenrolle `IS_SP_CONTENT_CONNECTION_CLASS` verwendet.

## 2.2 Wichtige Schnittstellen

### 2.2.1 BSP Client Klasse

#### 2.2.1.1 IF\_SRM\_SP\_CLIENT\_BSP

In der Methode `GET_BSP_APPLICATION` publizieren Sie den Namen der BSP-Applikation sowie des Controllers.

Beispiel:

```
method IF_SRM_SP_CLIENT_BSP~GET_BSP_APPLICATION .
    bsp_application-appname = 'SRM_NOTE'.
    bsp_application-controllername = 'NOTE.DO'.
endmethod.
```

#### 2.2.1.2 IF\_SRM\_SP\_ACTIVITIES

-siehe Dokumentation zur Service Provider Entwicklung-

### 2.2.1.3 IF\_SRM\_SP\_AUTHORIZATION

-siehe Dokumentation zur Service Provider Entwicklung-

## 2.2.2 BSP Query Klasse

### 2.2.2.1 IF\_SRM\_SP\_QUERY\_BSP

Analog zu IF\_SRM\_SP\_CLIENT\_BSP

## 2.2.3 BSP Controller Klasse

### 2.2.3.1 IF\_SRM\_SP\_BSP\_CONTROLLER

#### 2.2.3.1.1 Methode ON\_DESTROY

Diese Methode wird aufgerufen, wenn der SP vom Framework entladen wird. An dieser Stelle sollten ggf. gesetzte Sperren wieder freigegeben werden, um ‚tote‘ Sperreinträge zu vermeiden.

Beispiel:

```
method IF_SRM_SP_BSP_CONTROLLER~ON_DESTROY.  
  data: my_client type ref to zcl_my_client_class.  
  
  my_client ?= me->if_srm_sp_bsp_controller~sp_client.  
  
  my_client->release_all_locks( ).  
  
endmethod.
```

#### 2.2.3.1.2 Methode ON\_NEW\_REQUEST

Diese Methode wird aufgerufen, wenn ein neuer RM-Request zur Bearbeitung ansteht. An dieser Stelle sollte der Client angewiesen werden, die Daten aus dem Backend zu laden und ggf. aufzubereiten. Der ACTIVITY\_STATE des RM-Request-Objektes ist wie folgt zu setzen:

- IF\_SRM\_REQUEST=>ACTIVITY\_FINISHED\_WITH\_OK  
Dieser Wert ist zu setzen, wenn die Aktivität vollständig beendet wurde und der Service Provider *nicht* mehr angezeigt werden soll
- IF\_SRM\_REQUEST=>ACTIVITY\_ONGOING  
Dieser Wert ist zu setzen, wenn die Aktivität begonnen wurde und der Service Provider anschließend angezeigt werden soll (Normalfall).
- IF\_SRM\_REQUEST=>ACTIVITY\_ERROR bzw.  
NO\_AUTHORIZATION  
Dieser Wert ist zu setzen, wenn Fehler aufgetreten sind.

*Wichtig: Der eigentliche Aufruf des Views zur Visualisierung erfolgt erst in der –redefinierten- do\_request Methode des Controllers!*

Beispiel:

```
method IF_SRM_SP_BSP_CONTROLLER~ON_NEW_REQUEST.  
  data: my_client type ref to zcl_my_client_class.  
  
  my_client ?= me->if_srm_sp_bsp_controller~sp_client.  
  
  case im_request->get_activity( ).  
    when IF_SRM_ACTIVITY_LIST=>DISPLAY.
```

```
        my_client->open( im_for_update = ' ' ).
    when IF_SRM_ACTIVITY_LIST=>MODIFY.
        my_client->open( im_for_update = 'X' ).
    endcase.

    im_request->set_activity_state( if_srm_request=>activity_ongoing ).

endmethod.
```

### ***2.2.3.1.3 Methode ON\_USER\_EVENT***

Diese Methode wird aufgerufen, wenn eine Benutzerinteraktion (z.B. Event von HTMLB-Element; Klick auf Button) stattgefunden hat. An dieser Stelle sollte die Benutzerinteraktion ausgewertet und verarbeitet werden. Soll der Service Provider anschließend nicht mehr angezeigt werden („Exit“ oder „Close“ Funktion), so ist der ACTIVITY\_STATE am RM -Request-Objekt auf IF\_SRM\_REQUEST=>ACTIVITY\_FINISHED\_WITH\_OK zu setzen. Anschließend ist über IF\_SRM\_BSP\_CLIENT\_EVENT~SEND\_ASYNC\_ANSWER die asynchrone Antwort zu verschicken.



Beispiel:

```
method IF_SRM_SP_BSP_CONTROLLER~ON_NEW_REQUEST.  
  data: my_client type ref to zcl_my_client_class.  
  
  my_client ?= me->if_srm_sp_bsp_controller~sp_client.  
  
  event = cl_htmlb_manager=>get_event( runtime->server->request ).  
  
CASE event->server_event.  
  
  WHEN 'saveClick'.  
    my_client->save( ).  
  
  WHEN 'exitClick'.  
    my_client->discard( ).  
    my_client->release_all_locks( ).  
  
ENDCASE.  
  
im_request->set_activity_state( if_srm_request=>activity_ongoing ).  
  
me->if_srm_sp_bsp_controller~sp_client_event->send_async_answer(  
    im_request ).  
  
endmethod.
```

#### ***2.2.3.1.4 Methode ON\_REQUEST\_ANSWER***

Diese Methode wird aufgerufen, um Antworten auf einen vom SP selbst versandten RM-Request (z.B. Anzeige eines Unterobjekts) zuzustellen.

#### ***2.2.3.1.5 Methode ON\_REFRESH***

Diese Methode wird derzeit nicht vom Framework verwendet; es muß lediglich ein leerer Methodenrumpf angelegt werden.

#### ***2.2.3.2 IF\_SRM\_BSP\_CLIENT\_EVENT***

Eine Referenz auf IF\_SRM\_BSP\_CLIENT\_EVENT existiert als Attribut SP\_CLIENT\_EVENT an IF\_SRM\_SP\_BSP\_CONTROLLER; dieses Attribut wird vom Framework gefüllt und kann vom SP lesend genutzt werden.

##### ***2.2.3.2.1 Methode SEND\_REQUEST***

Diese Methode wird vom SP aufgerufen, um einen eigenen RM -Request zu versenden.

##### ***2.2.3.2.2 Methode SEND\_ASYNC\_ANSWER***

Diese Methode wird vom SP aufgerufen, um eine asynchrone Antwort zu versenden (z.B. bei Beendigung des Request).

## **3 Verwendung des BSP Framework**

### ***3.1 Das <srml:element> Tag***

Mit Hilfe des <srml:element> Tags können die Visualisierungen von SPs in eigene BSP-Applikationen eingebunden werden.

*Hinweis: Lesen Sie auch den F1 -Hilfetext zum <srmlb:element> Tag, um Informationen über die Verwendung des Tags zu erhalten.*

## **3.2 Beispielprogramme**

Ein Beispiel zur Verwendung des <srmlb:element> Tags finden Sie in der BSP Applikation SRM\_DEMO\_RECORD.

## **3.3 Attribute**

### **3.3.1 ID**

Das Element muß mit einer eindeutigen ID versehen werden, um die richtige Zuordnung des zentral verwalteten Zustands zu ermöglichen. Die ID muß eindeutig innerhalb des internen Modus sein, d.h. pro Tag und Benutzer-Login muß eine ID vergeben werden.

### **3.3.2 EMPTY\_CONTENT\_TEXT**

Hier kann ein Text angegeben werden, der angezeigt werden soll, wenn kein Element im Tag geladen ist.

### **3.3.3 EMBEDDED**

Ist der Schalter EMBEDDED auf IF\_SRM=>TRUE gesetzt, so ist die einbettende BSP -Applikation für das Rendern der HTMLB -Tags <htmlb:content> und <htmlb:page> selbst verantwortlich. Dies ist dann sinnvoll, wenn das Tag innerhalb einer BSP -Seite eingesetzt wird, auf der sich neben Records Management noch weitere Forms bzw. Elemente befinden.

### **3.3.4 EVENT\_CALLBACK**

Hier kann eine Instanz einer Klasse übergeben werden, deren Methoden vom RM Framework aufgerufen werden, wenn das dargestellte Element seinerseits einen RM-Request auslöst (z.B. wenn in einer Akte ein Dokument ausgewählt wird). Die Klasse muß das Interface IF\_SRM\_BSP\_CALLBACK implementieren.

## **4 Ausgelieferte BSP Service Provider**

### **4.1 Organizer**

Der ausgelieferte BSP Service Provider „Organizer“ ermöglicht die Anzeige eines mit der Transaktion SRMVIEWGEN erstellten View für BSP. Über den View kann der Benutzer –je nach Konfiguration - die Recherchefunktionen sowie direkt einzelne Objekte erreichen.

Der BSP Organizer ermittelt die dem Benutzer zugeordneten Rollen, und sucht innerhalb dieser Rollen nach zugeordneten Organizer Views.

*Hinweis: für die Verwendung des BSP Organizers muss dem Benutzer **genau** ein BSP View über die zugeordnete Rolle (PCFG) zugeordnet sein, d.h. es*

sind weder mehrere BSP Views zulässig, noch existiert die in SAPGUI vorhandene Standardsicht.

## **4.2 Akte**

Der ausgelieferte BSP Service Provider für Akte ermöglicht das Anzeigen und Ändern von Akten, sowie das Einfügen und Löschen von Elementen in der Akte.

## **4.3 Dokumente**

Der ausgelieferte BSP Service Provider für Dokumente ermöglicht das Anzeigen sowie den Upload von Dokumenten.

## **4.4 Archivdokumente**

Der ausgelieferte BSP Service Provider für Dokumente ermöglicht das Anzeigen sowie den Upload von Archive Link Dokumenten.

## **4.5 Notizen**

Der ausgelieferte BSP Service Provider für Notizen ermöglicht das Anzeigen von Notizen.

## **4.6 URL**

Der ausgelieferte BSP Service Provider für URLs ermöglicht das Anzeigen von URLs.

## **4.7 Suchdialog für Akten, Dokumente und Notizen**

Über den Standard-Recherchedialog ist die Suche nach Akten, Dokumenten und Notizen möglich. Der Standard-Recherchedialog kann durch die Verwendung einer Extension an die Kundenbedürfnisse angepasst werden.

## **4.8 Implementieren einer Extension-Klasse**

Eine Extension-Klasse muß das Interface IF\_SRM\_GENSP\_QUERY\_EXT implementieren.

### **4.8.1 Methode GET\_ICON**

Liefert das Icon, das in der Suchliste dargestellt wird. Die Standardimplementierung liefert das Icon, das für den jeweiligen Service Provider registriert ist, zurück.

### **4.8.2 Methode GET\_PARA\_BEE**

Über diese Methode besteht die Möglichkeit, die Darstellung der Suchmaske zu beeinflussen. Es muß eine Referenz auf eine sog. BSP Extension Expression (BEE) zurückgegeben werden (Referenz auf IF\_BSP\_BEE).

### **4.8.3 Methode GET\_PARA\_TAB**

Wird eine benutzerdefinierte Suchmaske verwendet, so muß in dieser Methode die Umsetzung der Werte aus den Eingabefeldern der benutzerdefinierten Suchmaske erfolgen. Die Werte der Eingabefelder können aus dem HTTP-Request-Objekt (IM\_REQUEST) dekodiert werden;

hieraus sind die Suchparametertabellen EX\_PROP\_QUERY und EX\_CONTENT\_QUERY zu versorgen. Über EX\_PROP\_LIST kann gesteuert werden, welche Attribute zu den gefundenen Einträgen geliefert werden.

*Hinweis: Nähere Informationen zum Aufbau der Parametertabellen finden Sie in der Entwicklerdokumentation zum Generic SP.*

#### **4.8.4 Methode GET\_RESULT\_FIELDCAT**

Liefert den Feldkatalog, der für die Darstellung der Suchergebnisse im HTMLB Table Control benötigt wird. Die Versorgung der Felder des Feldkataloges erfolgt in der Methode RENDER\_CELL.

#### **4.8.5 Methode GET\_TITLE**

Liefert den Titel des Recherchedialogs.

#### **4.8.6 Methode RENDER\_CELL**

In dieser Methode findet die Darstellung der einzelnen Felder des Suchergebnisses statt. Sie wird für jedes Feld jeder Zeile des Suchergebnisses einmal aufgerufen. Über den Importing -Parameter IM\_COLUMN\_ID kann die dargestellte Spalte, über die Importing -Parameter IM\_DOCUMENT\_DATA und IM\_DOCUMENT\_PROP die dargestellte Zeile bzw. deren Attribute ermittelt werden. Über den Returning -Parameter RE\_REPLACEMENT\_BEE muß eine Referenz auf eine BSP Extension Expression (IF\_HTMLB\_BEE) zurückgegeben werden.

*Hinweis: Weitere Informationen über BSP Extension Expressions und deren Verwendung im Rahmen des HTMLB Table Control finden Sie in der Online - Dokumentation unter dem Stichwort „Tableview Iterator“.*

### **4.9 Verwenden einer Extension-Klasse**

Die Extension-Klasse wird in der Transaktion SRMREGEDIT einer Elementart (nicht einem Service Provider!) zugeordnet. Öffnen Sie hierzu die Elementart im Änderungsmodus, und geben Sie als Connection Parameter „GENSP\_QUERY\_EXT“ den Namen der Extension-Klasse an. Ist zu einer Elementart keine Query Extension definiert, so wird die Standardklasse CL\_SRM\_DEFAULT\_QUERY\_EXT verwendet.

## **5 Verwendung der BSP Service Provider im Portal**

Für die Verwendung der BSP Service Provider im Portal muß zunächst eine BSP-Applikation implementiert werden, die das <srn:element> Tag einbindet. Diese BSP -Applikation kann dann über die Administrationsfunktion des Portals als IView eingebunden werden. Das <srn:element> Tag kann bei Bedarf zusammen mit Javascript -Coding zum Versand und Empfang von Portal-Events benutzt werden; in diesem Fall ist das benötigte Javascript auf der BSP Seite, die auch das <srn:element> Tag trägt, einzubinden. Durch Verwendung der entsprechenden BSP -Funktionalität können eingehende Portalevents umgesetzt und an das Tag weitergeleitet werden; sollen Ereignisse aus dem Tag bzw. aus Records Management in Portal -Events

umgesetzt werden, so kann hierfür die EVENT\_CALLBACK Funktion des <srm:element> Tags benutzt werden.

Eine Beispielimplementierung für einen Portalservice ist die Seite iview.htm der BSP-Applikation SRM\_DEMO\_RECORD. Sie ermöglicht eine Suche nach Akten bzw. Dokumenten sowie deren anschliessende Darstellung und kann direkt verwendet oder als Vorlage für eigene Implementierungen genutzt werden. Wird die Seite direkt verwendet, so können die Parameter RMS\_ID und SPS\_ID genutzt werden, um die Elementart und das RMS auszuwählen, mit dem die Suche arbeiten soll.

*Hinweis für Portalnutzer: Soll auf einer Portalseite mehr als eine Instanz des <srm:element> Tags genutzt werden, so muss unbedingt darauf geachtet werden, dass der Parameter ID des jeweiligen Tags eindeutig ist, z.B. durch Verwendung einer GUID. Ansonsten kann das RM BSP Framework keine eindeutige Zuordnung zwischen Tag und Service Providern vornehmen.*

Weitere Hinweise zur Benutzung von BSP Seiten im Portal finden Sie in der Portal-Dokumentation sowie in der Dokumentation zu Business Server Pages.